



FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.

1300 I STREET, N.W.
WASHINGTON, DC 20005-3315

202 • 408 • 4000
FACSIMILE 202 • 408 • 4400

ATLANTA
404•653•6400
PALO ALTO
650•849•6600

WRITER'S DIRECT DIAL NUMBER:

TOKYO
011•813•3431•6943
BRUSSELS
011•322•646•0353

JC682 U.S. PTO
09/540401
03/31/00

(202) 408-4000

March 31, 2000

ATTORNEY DOCKET NO. 07442.0010-02

Box PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

Re: Continuation-in-part U.S. Patent Application
Title: METHODS AND APPARATUS FOR COMPUTER
TRAINING RELATING TO DEVICES USING A RESOURCE
CONTROL MODULE
Inventors: James L. Boney, Terrance C. Slattery, and Shawn G. Coville

Sir:

We enclose the following papers for filing in the United States Patent and Trademark Office in connection with the above patent application.

1. Application - 40 pages, including one cover page, 4 independent claims, and 4 claims total.
2. Drawings - 33 sheets of informal drawings (Figures 1-24).
3. Checks totaling \$768 for the filing fee and one additional independent claim.

This application is being filed under the provisions of 37 C.F.R. § 1.53(b) and (f). Applicants await notification from the Patent and Trademark Office of the time set for filing the Declaration.

Please accord this application a serial number and filing date.

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.

Assistant Commissioner of Patents
March 31, 2000
Page 2

The Commissioner is hereby authorized to charge any additional filing fees due and any other fees due under 37 C.F.R. § 1.16 or § 1.17 during the pendency of this application to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

By

Charles W. Chesney
Reg. No. 45,874

CWC/wmr
Enclosures

United States Patent Application

of

James L. Boney

Terrance C. Slattery

Shawn G. Coville

for

Methods and Apparatus for
Computer Training
Relating to Devices Using
a Resource Control Module

RELATED APPLICATIONS

This application is a continuation-in-part of U.S. Patent Application Serial No. 09/365,243 filed July 30, 1999, which is hereby incorporated by reference.

In addition, the following applications contain subject matter related to the subject matter of the present application, are assigned to the assignee hereof, and are hereby incorporated by reference.

1. Terrance C. Slattery, Shawn G. Coville, Gregory R. Long, M. Tylor Burton III, "Methods and Apparatus for a Virtual Classroom." U.S. Patent Application Serial No. 09/364,047, filed July 30, 1999.
2. Terrance C. Slattery, Shawn G. Coville, Gregory R. Long, M. Tylor Burton, Lawrence F. Galvin, "Methods and Apparatus for Computer Based Assessment Relating to Devices." U.S. Patent Application Serial No. 09/364,048, filed July 30, 1999.
3. Terrance C. Slattery, Shawn G. Coville, Gregory R. Long, Karl J. Schmidt, Kenneth C. M. Stevenson III, Mark W. Decker, "Methods and Apparatus for a Learning Structure in Computer Based Training Relating to Devices." U.S. Patent Application Serial No. 09/364,046, filed July 30, 1999.
4. Terrance C. Slattery, Shawn G. Coville, Karl J. Schmidt, Kenneth C. M. Stevenson III, "Methods and Apparatus for a User Interface for Computer Based

Training Relating to Devices." U.S. Patent Application Serial No. 09/364,045, filed July 30, 1999.

BACKGROUND OF THE INVENTION

The present invention relates to computer based training, and more particularly, to methods and apparatus for computer based training providing hands-on training related to devices.

Traditionally, students have been taught about devices in a classroom setting where the students were lectured about the devices and presented with written tests. Although students have been taught in a lab setting, this requires students to travel to a specific location to perform the lab. Thus, making it difficult and inconvenient for students without easy access to a lab.

Thus, it has been difficult for students to gain hands-on experience regarding devices in a classroom setting, in their office place, or at home. Further, the need for more hands on time with devices is becoming critical for professionals that are working towards certifications such as the CISCO Certified Internetworking Expert (CCIE) certification, which is the highest certification level bestowed by CISCO.

Accordingly, there is a need for a distance based solution that provides students with hands-on training with devices.

SUMMARY OF THE INVENTION

Accordingly, the present invention is directed to methods and apparatus that substantially obviates the above and other disadvantages of the prior art.

Methods and systems consistent with the present invention include a system for training a user regarding controlling a device. This system includes a user computer for accepting device control information regarding controlling the device for the purposes of training, and a device controller remotely connected to the user computer. The device controller preferably includes a processor acting as a proxy for connecting the user computer to the device means for receiving the device control information from the user, and means for transferring the device control information to the device.

In another aspect, the invention comprises a method for training a user regarding controlling a device. This method includes a user remotely connecting to a proxy in a device controller using a user computer; the user computer transferring user information regarding control of the device to the proxy, and the proxy transferring the user information to the device.

In yet another aspect, the invention comprises a system for training a user regarding controlling a device, including a user computer for accepting device control information regarding controlling the device for the purposes of training, a server/controller including a processor for transferring overhead information for performing overhead functions relating to the device using a generic set of commands, and a device controller that includes a processor for receiving the overhead information from the server/controller, exercising control over the device using the

overhead information, receiving the device control information from the user, and transferring the device control information to the device.

In yet another aspect, the invention comprises a method for training a user regarding controlling a device, including the steps of a server controller transferring overhead information using a generic set of commands to a device controller in order to perform overhead functions relating to the device, the device controller using the overhead information to exercise control over the device for the purposes of placing the device in the initial configuration, a user remotely connecting to the device controller using a user computer, the user computer transferring device control information regarding control of the device to the device controller, and the device controller transferring the device control information to the device.

The summary of the invention and the following detailed description should not restrict the scope of the claimed invention. Both provide examples and explanations to enable others to practice the invention. The accompanying drawings, which form part of the description for carrying out the best mode of the invention, show several embodiments of the invention, and together with the description, explain the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the Figures:

Figure 1 illustrates a block diagram of a computer-based system for training of devices in accordance with methods and systems consistent with the invention;

Figure 2 illustrates a block diagram showing a distributed set of pod controllers, in accordance with methods and systems consistent with the invention;

Figure 3 provides a detailed diagram of a pod controller, in accordance with methods and systems consistent with the invention;

Figure 4 illustrates a graphical interface, in accordance with methods and systems consistent with the invention;

Figure 5 provides a flow diagram for a computer based training system, in accordance with methods and systems consistent with the invention;

Figure 6 illustrates an example page that may be displayed to a user to log in or to request account information, in accordance with methods and systems consistent with the invention;

Figure 7 illustrates a page that may be used for setting up a user account, in accordance with methods and systems consistent with the invention;

Figure 8 illustrates Relay Control Boxes, in accordance with methods and systems consistent with the invention.

Figure 9 illustrates a block diagram of a pod controller including a wiretap, in accordance with methods and systems consistent with the invention;

Figure 10 provides an illustration of a Virtual Classroom, in accordance with methods and systems consistent with the invention;

Figure 11 provides a flow diagram for the virtual classroom, in accordance with methods and systems consistent with the invention;

Figure 12 illustrates a logical diagram of a skills learning module, in accordance with methods and systems consistent with the invention;

Figure 13 provides a logical description of information that may be stored by the assignment and story sub-module, in accordance with methods and systems consistent with the invention;

Figure 14 illustrates an example of an assignment and story sub-module, in accordance with methods and systems consistent with the invention;

Figure 15 illustrates an example of a page that provides a user with a suggested approach, in accordance with methods and systems consistent with the invention;

Figure 16 illustrates a template for a page that provides a user with a sample solution, in accordance with methods and systems consistent with the invention;

Figure 17 illustrates an example of an html page that provides a sample solution, in accordance with methods and systems consistent with the invention;

Figure 18 illustrates an example of a page for the check results module, in accordance with methods and systems consistent with the invention;

Figure 19 illustrates an example of a page for the check results module, in accordance with methods and systems consistent with the invention.

Figure 20 illustrates an example of a page that includes archived information for a lab, in accordance with methods and systems consistent with the invention.

Figure 21 provides an illustration of an assessment engine that uses a wiretap to check the configuration of the devices, in accordance with methods and systems consistent with the invention;

Figure 22 illustrates an embodiment in which the server and controller contains an assessment engine, in accordance with methods and systems consistent with the invention.

Figure 23 illustrates a block diagram of a resource control module, in accordance with methods and systems consistent with the invention; and

Figure 24 illustrates a logical block diagram of the proxy module of a resource control module, in accordance with methods and systems consistent with the invention.

DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Engine

Figure 1 illustrates a simple block diagram of a computer-based system for training relating to devices. As shown, the system preferably includes customer premise equipment 12, a communications link 14, a firewall 16, a communications link 18, a server and controller 20, a database 22, a pod controller 24, and a pod 26. The Customer Premise Equipment (CPE) 12

preferably includes a computer 28 provided with a browser program 30 and a network application program 32. The browser 30 is preferably a browser for Internet/Intranet communications, such as a Netscape Navigator™ browser or a Microsoft Internet Explorer™ browser. The network application program 32 may be a program such as TELNET. The communications link 14 preferably traverses the Internet or an Intra-net. The pod controller 24 may control one or more pods 26 each of which may contain one or more user devices 40_1 to 40_3. In one embodiment, the user devices 40_1 to 40_3 are network equipment, such as CISCO type switches or routers. In other embodiments, these user devices may be Programmable Logic Controllers (PLCs), Chemistry Equipment, or any other type of device. Further, a pod controller 24 may also control one or more infrastructure devices (not shown). These infrastructure devices provide an authentic environment for which a real world scenario may be written.

Figure 2, illustrates a simple block diagram of an embodiment with multiple pod controllers 24_1 to 24_3. In this embodiment, pod controllers 24_1 and 24_2 are behind firewall 16_1, and pod controller 24_3 is behind firewall 16_2.

Figure 3 provides a more detailed diagram of the pod controller 24, consistent with an embodiment of the invention. The pod controller 24 may include a device control module 302, a user communications module 304, a mentor communications module 306, an infrastructure control module 308, a device communications, control, and multiplexor module 310, and an interface to device module 312. The pod controller 24 may be connected to user devices 314, and infrastructure devices 316.

The device control module 302 is used to control user accessible devices. It incorporates the control software that enables the pod control system to load starting configurations into the user devices, reset the user devices, and save final configurations. The control within this module preferably is high level and generic across all devices, increasing the modularity and maintainability of the overall system.

The user communications module 304 operates such that when a user connects to a user device, the connection is made through the user communications module. This module receives the connection from the network and validates the user's authorization to access specific devices. This module further translates information received by a user in one protocol into a protocol for feeding into the user device.

The mentor communications module 306 permits a mentor to monitor and participate in controlling the user devices during a learning exercise. The mentor communications module 306 authenticates and authorizes the mentor to connect to specific devices through a computer. A mentor may be a program, such as an Artificial Intelligence program, a person, or any type of hardware or software capable of aiding a user in learning about the user device and its operation.

The infrastructure control module 308 allows additional devices to be interconnected to the user devices in order to replicate real-world scenarios. These devices are part of the infrastructure and require separate control by the pod controller. As such, this module provides the control of the infrastructure devices that are needed to create a real-world scenario for the user. The infrastructure devices 316 are discussed in greater detail below. Further, this module, in conjunction with device, communications, control and multiplexer modules 310, permits

devices to be moved between pods connected to the pod controller 24. This will be discussed in further detail below.

A device communications, control, and multiplexor module 310 provides low-level communications and control for each device. In addition, this module provides a mechanism for multiple modules to simultaneously communicate with a single device. Further, as discussed above, this module permits devices to be moved between pods connected to the pod controller 24. For example, Figure 2 illustrates user device 40_4 as part of pod 26_2. If a user wishes to perform an assignment that needs four user devices, the device communications, control, and multiplexor module 310 for user device 40_4 may be reconfigured to move user device 40_4 into pod 26_1. Further, the infrastructure control module 308 may also be reconfigured so that the infrastructure devices necessary for supporting this move are also reconfigured. Thus, pod 26_1 would now include four user devices so that a user wishing to perform the assignment can use pod 26_1 and pod controller 24_1.

The interface to device module 312 is generally implemented in a combination of hardware and software, and provides the pod controller 24 with the capability of the pod controller 24 to communicate directly with the device. For example, if the device 314 uses an RS-232 or similar interface, this module may include software and hardware for implementing an RS-232 connection to the device. Other connection types are possible, such as Ethernet, and the connection type will be dependent on the specifics of the device. Further, this module is capable of receiving information from a user in an application layer protocol, such as TELNET, and translating it so that the information can be communicated directly to a user device.

Each of the modules may be implemented by software running on a processor or computer. As will be obvious to one skilled in the art, all of these modules may be implemented on a single processor, or multiple processors may be used. A variety of computer programming languages may be used to implement the software programs.

The infrastructure devices 316 permits user devices 314 to be interconnected into realistic configurations. The infrastructure devices are not directly accessible by the user. However, the system may implement functions that affect the configuration of these devices depending upon the user's actions. For example, a Frame Relay switch may be used to implement a wide area network connection between user devices. Such a switch may need to be configured and controlled for specific exercises, but because multiple users may need to share it, the switch will not be directly available to the user.

The server and controller 20 may be a computer including memory, a processor, a communications port, a display screen, keyboard, and a mouse. Further, the server and controller 20, for example, may be a web server, such as an Apache or an AOL Server, that uses the Common Gateway Interface (CGI) to allow programs to be executed within the server and controller 20. These programs may provide services to a user via the web such as static web pages, a scheduling mechanism to allow a user to schedule the execution of a specific lab or course at a given time, a lab or course catalog displayed via a web page showing all labs or courses which can be executed, an electronic commerce engine through which users may purchase labs or courses online, and/or a web page which provides a graphical interface for accessing the user devices in specific labs and initiating device control software.

The server and controller 20 may directly provide static web pages or they may be dynamically created from templates and other data contained within a system database. A scheduling component may interface to a database of reservations to determine the availability of devices for a lab. Further, the scheduling component may have the capability to add, delete, or change a user's schedule. A course catalog may include a nearly static page generated from a database of existing labs or courses. This page may be displayed to a user as an html web page. Further, the server and controller 20 may connect to one or more systems, such as electronic commerce systems, training/student management systems, personnel records systems, etc. This connection may be over an Internet, or such systems may reside within the server and controller 20. An electronic commerce system is a system for conducting commerce such as on-line ordering and accepting credit payments over an Internet.

The server and controller 20 may send either static or dynamic web pages to the user's CPE 12 so that they may be displayed to the user through their browser. The content of the user interface page may contain buttons and hot links for the user to invoke the device-specific operations necessary to control the user devices for the purposes of accomplishing a lab or course.

The user computer 28 may be a standard user computer such as an IBM compatible Personal Computer (PC) or Macintosh. Further, the user computer may include a memory, a processor, a keyboard, a mouse, and a port for receiving and transmitting information. The keyboard and mouse may be used by a user to enter information into the computer. The memory may be used for storing programs, such as the browser and network applications programs. The

processor may be used for executing these programs. The display screen may be used for displaying information to the user such as html pages, and screens for the browser and network applications programs. The communications port may be used for sending and receiving information over a communications link.

Figure 5 provides a flow diagram for the computer based training system of Figure 1. A user initializes the system by instructing the browser 30 on his/her computer 28 to connect to the server and controller 20 (S502). The firewall 16 is preferably set up to allow the browser 30 and server and controller 20 to freely communicate. The server and controller 20 preferably displays a page to the user requesting an account identification and password and then uses this information to determine if the user has an account (S504). Figure 6 illustrates an example html page 600 that may be displayed to a user to request a user's account identification 602 and password 604. The user can then click on the submit button 606 to submit the information once it has been entered. The user account may contain, among other things, a course for the user and course specific information. If the user has an account, the course is selected and started (S506). If the user does not have an account, he/she is directed to a page for setting up a user account (S518). For example, as illustrated in Figure 6, a user without an account identification is directed to click on a Register button 608. Figure 7 illustrates an html page 700 that may be used for setting up a user account. This page may request, for example, his/her name 702, address 704, a user name 706, and a password 708.

After the course is selected, the server and controller 20 sends an instruction to the pod controller to initialize a pod 26 associated with the selected course (S508). Further, the server

and controller 20 sends an instruction to the firewall 16 to permit the user's CPE 12 to access the pod controller 24 (S510). The user then connects to the pod controller 24 using a network application program 32, such as Telnet (S512). Through this mechanism, the user can exercise control over the user devices and perform the assigned training exercise.

A graphical interface may be presented to the user that includes buttons and/or hot links regarding the various user devices. These buttons and/or hotlinks may be used by the user to perform various functions regarding the user devices. The graphical interface may include a button for resetting a user device to its initial starting configuration, a button for saving an existing configuration, a button for loading a new configuration, or a button for forcing the user device into a diagnostic control mode. The user may perform these functions by clicking on a specifically labeled button on the graphical interface. The specific buttons or hotlinks and their respective functions will depend on the specifics of the user devices that make up the lab or course. For example, a chemistry lab may use buttons for controlling a mechanical shuttle which moves a beaker to different locations within an experiment, the addition of a chemical to the beaker, controlling a heating element, or controlling the agitation of the solution within the beaker.

Figure 4 provides an example of a graphical interface a user may see when interacting with a lab including three Cisco routers. The available functions in this example are: reset 402, save configuration 404, load configuration 406, and cycle-break 408. The user may click on the icon 410 of a user device to open a communications connection to the user device when completing the objectives of the lab or course. Further, in this example, the user may click on links for the

instructions 412, scenario and assignment 414, mentoring 416, view plan 418, suggested approach 420, sample solution 422, and check results 424. By clicking on one of these links, the user will be taken to an associated web page. These links will be discussed in greater detail later.

After completion of the course, the user's CPE 12 is disconnected from the user devices 40_1 to 40_3 by the pod controller 24, and the user devices are reset to their original settings (S516). For routers, these user devices can be reset by sending a specific sequence to the router that permits the system to gain access to the router even if the user alters the passwords for the router. This may be accomplished by turning off the router, and then turning it back on. During the boot up, control characters can be sent to the router to cause it to enter ROM mode. Once the router is in ROM mode, the user device can be reset so that another user can use the user devices regardless of what the previous user did to the user device.

If the user device is a CISCO-type switch, a specific button needs to be pushed during power up in order to enter ROM mode. This switch may be replaced by a relay to close the switch. Thus, the pod controller could reset this user device by closing the relay, or by manually operating the switch. As an example, Figure 8 illustrates three Relay Control Boxes 802, 804, and 806 for resetting devices in which a physical switch needs to be pressed at a particular point in time. Each Relay Control Box may include eight internal relays for controlling eight independent devices. Each relay replaces the physical switch in the device. As illustrated, a master Relay Control Box 802 is connected directly to the Pod Controller 24, and each client Relay Control Box 804 and 806 is serially connected to the master Relay Control Box 802. When a device is to be reset, the pod controller 24 sends an address for the device to the Relay Control

Boxes 802, 804, and 806. The Relay Control Boxes 802, 804, and 806 then use this address to close a relay associated with this address that in effect closes the switch for the device. Thus, the Relay Control Boxes and Pod Controller can individually open and close the switches for the particular devices. This example is provided as merely one possible implementation, and as will be obvious to one of skill in the art numerous other implementations are possible. Further, the specifics of how to reset a device will depend on the particulars of the specific device.

In one embodiment, user devices 40_1 to 40_3 are CISCO-type routers and are connected to the pod controller through a COM port. In this embodiment, the pod controller 24 converts the user information from the application layer format it is received in (for example, TELNET) to a format that can be sent to the router through the router's COM port. This is preferably accomplished by the user communications module 304 of the pod controller 24.

Figure 9 illustrates a block diagram of a pod controller 24 including a wiretap 902, according to an embodiment. As shown, the pod controller of this embodiment is connected to a user's network application program 32, such as TELNET, over a network connection 904 that preferably includes, referring back to Figure 1, a communications link 14 from the CPE 12 to the firewall 16 and a connection from the firewall 16 to the pod controller 24 over communications link 18. Also connected to the wiretap 902 is a mentor's equipment 906 operating a network application 908, such as Telnet. Through this mechanism, a mentor can monitor the instructions the user is sending to the user devices in the pod. As such, the mentor can "watch" what the user is doing, so that the mentor can ensure that the user is properly controlling the user device. Various software and hardware can be used to implement the wiretap. For example, KIBITZ, a

program available with the freeware package EXPECT, allows two users to collaborate over a network while interacting with a single program. Thus, by using one KIBITZ for each user device, everything the user types can be seen by the mentor, and visa versa. Other programs with similar functionality may also be used to achieve this type of operation. The wiretap 902 may be implemented in the device communications, control, and multiplexor module 310 of the pod controller 24.

Additionally, this wiretap may permit the mentor to take control of the user devices in the pod so that the user can "watch" the instructions the monitor is sending to the user devices. Thus, the mentor can show the user how to correctly operate the user devices to perform a task. In another embodiment, the wiretap may be connected to an Artificial Intelligence program that oversees the user's operation of the user devices.

Virtual Classroom

Figure 10 provides an illustration of a Virtual Classroom, 1000 consistent with an embodiment of the invention. As shown, the Virtual Classroom 1000 preferably includes a plurality of student terminals 1010_1 to 1010_16, an instructor terminal 1012, a communications link 1014, a firewall 1018, a server and controller 1020, a pod controller 1024, and a pod 1026. The pod 1026 preferably includes a plurality of user devices 1040_1 to 1040_3. The number of devices needed will depend on the class objectives. The server and controller is optionally connected to a database 1022 that stores class specific information. The student terminals 1010_1 to 1010_16 each preferably include a browser (not shown) that may be a browser for

Internet/Intranet communications, such as a Netscape Navigator™ browser or a Microsoft Internet Explorer™ browser. Further, each student terminal 1010_1 to 1010_16 preferably includes a network application (not shown) for sending control information to the user devices. In one embodiment, the network application program is TELNET. The communications link can be any type of communications link, and may be, for example, a link across an Internet or Intranet. Further, a server (not shown) may be used to connect the student and instructor terminals to an Internet or Intra-net that links the terminals to the server and controller 1020 and pod controller 1024.

Individual students need not be located at the same physical location. For example, one student may be located at one physical location, another student may be located in a physical location in a different city, a third student may be located in yet another city, and so on. The communications between the instructor and students can be supported using a variety of communications applications. For example, ILINC, Placeware, or Caliver communications applications may be used for supporting these communications. Further, the class can be conducted synchronously or asynchronously.

Figure 11 provides a flow diagram for the virtual classroom, consistent with the invention. An instructor initializes the class session by sending information from the instructor terminal 1012 to the server and controller 1020 (S1102). The server and controller 1020 then sends initialization information to the pod controller 1024 (S1104). The pod controller then resets the user devices 1040_1 to 1040_3 (S1106). The server and controller 1020 then instructs the firewall 1018 to permit the student terminals to access the pod controller 1024 (S1108). Through

this system and mechanism students can be trained regarding controlling a user device by exercising actual control over the user devices in the pod (S1110).

After the class ends, the instructor can instruct the server and controller 1020 to save the configurations of the user devices 1040_1 to 1040_3 to allow the class to pick up where it left off at a later time. The user devices are then preferably reset to permit another class to use the user devices (S1112).

Organization of Learning Structure

Traditionally, students are taught using a learning structure that involves an instructor first lecturing or telling students about a subject, such as how to operate a device. Next, the instructor shows the students how to perform the task, and finally students are provided with a task and asked to perform it.

In an embodiment of the invention, rather than first instructing the students regarding the operation of a device, students are invited to try and perform a task regarding the device. If the student is unable to perform the task they are provided with a first level of help. This first level of help includes clues or hints regarding performing the task and top-level information. If the student needs further help, he/she is provided with a second level of information. This second level of information includes detailed information regarding performance of the task, and in one embodiment includes showing the user the exact steps to take in order to perform the task.

Figure 12 illustrates a logical diagram of a skills learning module 1200 consistent with the invention. The skills learning module 1200 may include module metadata 1210, an assignment

and story module 1220, a suggested approach module 1230, a sample solution module 1240, and a check results module 1250. In an embodiment, the skills learning module is stored in database 22 along with each of its sub-modules. Each of these sub-modules may be presented to a user in the form of a web page in an html format. By structuring the data in a hierarchical manner, the user may view only that information which is needed at a particular time.

Module metadata 1210 is a set of data that describes the learning module, and may include information regarding the learning environment for the module, the author of the module, the difficulty level, target audience, and other like information. For example, module metadata 1210 may include the assignment's name, its difficulty, the expected time to complete it, its authorship, and the skills or knowledge that it exercises.

The assignment and story sub-module 1220 stores information regarding the context of an assignment so that a user may frame the learning experience in the real world. Figure 13 provides a more detailed logical description of information that may be stored by the assignment and story sub-module 1220. As illustrated the assignment and story sub-module 1220 may include a details sub-module 1310, an assignment sub-module 1320, a story sub-module 1330, a conditions sub-module 1340, a notes sub-module 1350, and a diagram sub-module 1360. Figure 14 illustrates an example of an assignment and story sub-module that may be presented to a user in the form of an html web page.

The details sub-module 1310 provides module metadata that is useful to the user. For example, as illustrated in Figure 14, the details sub-module may include data regarding the title of the assignment, the technology involved, the level of difficulty, the amount of time to complete

the task, the certification it may be used for, the desired learner outcome, and the desired network outcomes.

The assignment sub-module 1320 provides a brief textual description of the environment in which the assignment takes place. For example, as illustrated in Figure 14, the assignment sub-module 1420 describes the assignment as designing an Appletalk numbering plan and enabling Appletalk routing.

The story sub-module 1330 provides a detailed textual description of the environment in which the assignment takes place. It typically mirrors a real-world scenario. For example, as illustrated in Figure 14, the story sub-module 1430 describes the story for the assignment as: "Your network manager has told you that your network will soon have to carry Appletalk traffic. In order for this to happen you must plan an Appletalk numbering scheme and assign Appletalk zone names for each of the segments in your network. You will also enable Appletalk routing on all of the active interfaces on your routers. Once Appletalk is enabled on the routers and configured on the interfaces, you should verify that Appletalk is functioning properly."

The conditions sub-module 1340 provides supporting information. For example, as illustrated in Figure 14, the conditions sub-module 1440 may include information such as: "IP routing is already up and running on this network. DO NOT CHANGE ANY OF THE IP ROUTING CONFIGURATIONS.

Your Apple administrator has given you the following range of Appletalk network numbers, 2000-2999. You may use any number with that range to assign a unique Appletalk network number to each segment in the network. All of the serial links should

be configured in the 'cereal zone.' You should make up unique zone names for each of the Ethernet interfaces."

The notes sub-module 1350 provides exceptions to real-world implementations or things that should be taken into consideration in the performance of the module. For example, as illustrated in Figure 14, the notes sub-module states that the exceptions to real world conditions for this assignment are that the serial links between routers are implemented via direct connections in this lab and do not actually connect through any leased line services for the serial links.

The diagram sub-module 1360 provides a picture or visual of the module's starting equipment. For example, as illustrated in Figure 14, the diagram sub-module 1460 illustrates three interconnected routers.

The suggested approach sub-module 1230 (Fig. 12) provides a high level description of an approach that the learner could take to accomplish the module's assignment. Figure 15 illustrates an example of a page that provides a user with a suggested approach. As illustrated, in this example, the suggested approach is for the user to: "Figure out the Appletalk numbering plan; assign a unique Appletalk cable range to each network segment; note the Appletalk zone names on each network; enable Appletalk routing on the routers; configure the appropriate Appletalk zone names on each active router interface; then verify proper Appletalk operation using show commands."

The sample solution sub-module 1240 (Fig. 12) contains a subject-matter expert's solution to the module's assignment. Figure 16 illustrates a template for a page that provides a user with a

sample solution. Each solution can be broken down into a series of tasks 1602. The initial display may only show each task, along with a one-line description for the task. The user may then click on a task to expose more detailed information regarding the tasks, such as hints 1604 and steps 1606. A hint 1604 is designed to assist the user in accomplishing the tasks and is useful for users who need some initial direction on how to get started with the task, but do not need detailed information on the process. A step 1606 is a one line description that may include an action to take, the result of that action, and an explanation on why the step was necessary and the result that was achieved. In another embodiment, the sample solution may show all the details of the sample solution. Figure 17 illustrates an example of an html page that provides such a sample solution.

The check results sub-module 1250, provides user assessment because during the course of performing a lab or course, a user may wish to check his or her final results. The user's results may be assessed in a variety of ways. For example, this assessment may be performed by examining the final configuration of the devices as compared to a final configuration produced by a subject matter expert. Figure 18 illustrates an example of a page that provides a user with one possible set of final device configurations which can be used to compare against the results obtained by the user. In another embodiment, the check results module may provide a set of steps that the user can execute to assess his or her final configuration. For example, in a Cisco router lab, the user may be told to how to examine the routing table of a specific device and look for specific outputs. Figure 19 illustrates an example of a page that provides a user with steps for checking their results.

Further, referring back to the example illustrated in Figure 4, the user may select from the graphical interface illustrated to view instructions regarding the lab 412. By clicking on this link, a page is displayed to the user regarding the general instructions for performing the lab. Also shown in the example illustrated in Figure 4 is a link to view plan 418. In this example, prior to beginning the lab, the user is presented with a series of predetermined questions designed to help the user structure his/her lab approach, thus allowing the user to plan his/her approach before beginning the lab. These questions may include, for example, questions regarding the user's plan of attack, the impact the user thinks his/her approach will have on the rest of the network and any potential trouble areas and anticipated challenges, and how he/she expects the lab to be helpful. Further, these questions may be predetermined questions or be adaptive questions that depend upon the user's answers to the questions. When the user later clicks on view plan 418, a web page is displayed to him/her that includes answers to these questions.

After a user completes a lab, in an embodiment, the user is led through a debrief question/answer section designed to reinforce the lessons taught by the lab and help the user identify ways that he/she might incorporate these skills in a production environment. The user's answers are then saved for future reference. These questions may include, for example, questions regarding how the user felt his/her plan worked out, the impact his/her approach might have had in a production environment, how he/she might change their approach, how he/she thinks his/her revised approach might work in a production environment, what he/she learned, and how the lab was helpful to him/her. Further, these questions may be predetermined questions or be adaptive questions that depend upon the user's answers to the questions.

In one embodiment, after completion of a lab, the final configurations of the user devices along with other information are archived so that the user may retrieve the information and configurations at a later time. Figure 20 illustrates an example of a page that includes archived results after a user completes a lab where the user devices are Cisco type routers. In this figure, the data is organized such that a user may select the component of the data they wish to view while hiding the details of the other components. Thus, a user may quickly select the section of data they wish to view and ignore the other sections. For example, as illustrated, a user may select to individually view plan 2002, debrief 2004, or saved configurations 2006. By selecting plan 2002, a user may view, for example, information regarding their answers to the questions regard their plan that they had previously entered. By selecting debrief 2004, a user may view information regarding their answers to questions during a debriefing session and by selecting saved configurations 2006, a user may view the final configurations of the user devices.

Assessment

Using the above described Engine or Virtual Classroom permits assessing a user's proficiency regarding a device by assessing the user/student's control over the device.

Traditionally, students are assessed based on taking a test where they are asked questions regarding the device or control of the device. In an embodiment of the present invention, rather than simply asking a student questions regarding the device, the student is presented with a real world task regarding the control of the device. The student controls the device to perform the task. After completing the task, the student is assessed on his/her performance.

Figure 21 provides an illustration of an assessment engine 2110 that uses a wiretap 2112 to check the configuration of the user devices 2114 once the assignment is completed. As with the wiretap discussed above, various configurations may be used to implement the wiretap 2112. For example, KIBITZ may be used to implement the wiretap 2112. The assessment engine 2110 may be a piece of software running on a processor or computer. Once the assignment is completed, the assessment engine 2110 may examine the configuration of the various user devices 2114 using the wiretap 2112. The assessment engine then checks this configuration against an expected configuration. It then may generate a score. Further, the assessment engine can inform the user of how their configuration differs from the expected configuration, and the potential consequences of these differences.

Figure 22 illustrates an embodiment in which the server and controller 2210 contains an assessment engine 2226. This assessment engine 2226 may be a piece of software running on a processor. At the completion of the assignment, the user's CPE 2212 may be disconnected from the user devices. The assessment engine 2226 then connects to the various user devices 40_1 to 40_3 through the wiretap 2228 of pod controller 24. The assessment engine 2226 then examines the configurations of the user devices 40_1 to 40_3 and compares these configurations with expected configurations. The assessment engine 2226 then may generate a score for the user, and provide him/her with feedback regarding the differences between his/her configurations and the expected configurations. Further, as will be obvious to one of skill in the art, in other embodiments, the assessment engine 2226 may connect to the various devices through a direct connection.

Further, as previously discussed in one embodiment a user may be presented with a set of steps that the user can execute to assess his or her final configuration. Referring back, Figure 19 illustrates an example of a page that provides a user with steps for checking their results.

Resource Control Module

In yet another embodiment, the pod controller is implemented by a Resource Control Module (RCM). The RCM provides a scalable and configurable mechanism for the remote control of and remote access to computing resources. Access to the RCM may be via the public Internet or an Intranet.

As with the pod controller, the user's interface to these devices is exactly the same as it would be in the real-world case of direct access to the actual devices in a functioning production environment. The RCM further permits the user's environment to be fully isolated, so that nothing the user can do will have any impact outside that isolated environment. The RCM can preserve and restore the devices to predefined and/or user-created system configurations on command. When the user is finished with the equipment, the RCM can quickly return the devices to a known stable state for use by another user, facilitating the sharing of one set of assets among many users, in a timely manner, with no manual intervention.

The RCM preferably operates independently of the particular server and controller used. For example, the above-described virtual classroom may use a different type of server and controller than a virtual lab type system (i.e., a single user exercising control over a pod, such as was described in the above Engine section). Because the RCM may be both physically and

logically separate, external applications may be insulated from the details of device control. This provides a flexible and scalable environment for integrating new devices and/or new application functionality.

Figure 23 illustrates a simplified block diagram of the RCM 2310 connected to a server and controller 2312, a user computer 2314, an instructor computer 2315, and a pod 2316 that includes one or more remote devices 2318. The server and controller 2312 may connect to the RCM 2310 over an Internet or Intranet 2330 or via a direct connection and may be a server controller for a virtual lab system (i.e., a system in which a single user exercises control over the pod devices), a virtual classroom, or any other type of server controller. The user computer 2314 may connect to the Resource Control Module over an Internet or Intranet 2318 or via a direct connection. The devices 2318 may include any type of device, such as a router, a switch, a network server, personal computers (PCs), UNIX workstations (SunTM, etc.), or any other type of device. Further, as with the previous described embodiments the connections between the server and controller 2312, user computer 2314 and RCM 2310 may be through a firewall (not shown).

The RCM 2310 preferably uses a component based architecture which includes a control module 2320, an operations module 2322, and a proxy module 2324 that perform separate independent functions. These software modules may be collocated on the same computing device (processor) or remotely located on separate computing devices. In addition, the RCM 2310 may include a RMI interface 2340, a database for storing overhead and status information 2342, a database for storing script names that are used in controlling devices 2344, and a file or database

for storing session logs 2346. Further, as will be obvious to one of skill in the art, all these databases may be combined in a single database.

In addition to including one or more devices 2318, the pod 2316 may also include a console router 2317, and infrastructure devices such as a power controller (not shown) for resetting power to the devices, and/or a relay control box for pressing any physical button on the device. The console router 2317 may be used for communicating with serial ports on a networking device and or for assigning a private or public IP address to a devices serial port. Further, the RCM 2310, like a pod controller, may also include device communications, control, and multiplexor modules (not shown) and interface to device modules (not shown).

The control module 2320 preferably accepts commands from an external application such as virtual lab or virtual classroom and communicates the commands to the appropriate module. This module allows for scalability in allowing new modules to be added to the system without modifying the current modules. The control module 2320 further provides a mechanism to physically and logically separate the RCM 2310 from the application environment.

The control module 2320 accomplishes this by employing a common set of instructions that different types of server and controller's 2312 can use to communicate with the control module 2320. These instructions are preferably in the form of a remote procedure call (RPC) language, such as JavaTM RMI. In an embodiment, the following general message types are used by the RCM's control module 2320:

addUser – Adds access for a user to a device

deleteUser – Removes user access from a device

changePerm – Changes user permission on a device

pause – Pauses telnet connection

resume – Resumes a paused telnet connection

startLog – Starts logging for a device

stopLog – Stops logging for a device

getLog – Retreives a log file

getUserList – Retreives all current users and their permissions

runOperation - performs an operation on the device such as SCRUB, SAVE,

LOAD, or CycleBreak. SCRUB returns the device to its original factory

settings; SAVE saves the current state of the device; and LOAD returns the

device to a previously saved state.

The control module 2320 preferably accepts requests via an RPC call, such as a JAVATM RMI call, from the server and controller 2312. Each call relates to a function within the control module 2320. Depending on the operation type, the control module either updates a database, such as database 2342, or simply passes the information to another module within the RCM 2310, such as the operations module 2322 or proxy module 2324. For script commands, the operation module 2322 spawns a script that may be stored in database 2344. The script then is used to perform a particular operation on the device 2318. The output of the script (pass or fail) is then passed back to the server and controller 2312 via the original RMI call. A more detailed description of a script is presented below.

The operations module 2322 preferably performs actual device control. This module is designed such that new devices can be added without major modification to the existing software. This is accomplished by encapsulating the device specific details in device specific scripts. The scripts may be written in a script language, such as Expect or PerlTM and are used to communicate with the device in a send this and expect that manner. Thus, by using a script, an entire procedure for a device can be scripted using basic device commands. The operations module 2322 by looking in database 2344 can tell from a particular instruction it receives, which script to call for each device. Thus, the operations module 2322 upon receiving a generic instruction to perform an operation on a device 2318 can use the particular script for the device 2318 to send the device specific instructions to the device 2318. In other words, the operations module 2322 may command the execution of the appropriate script based on generic information that is received from the control module 2320. In addition to device control, the operations module 2322 maintains the status of all pods and the devices within each pod.

The proxy module 2324 couples the user to the actual remote devices 2318 . The proxy module 2324 preferably provides authentication of the remote user prior to granting access to the remote device. Once granted, access is further controlled based on the application assigned user level (typically Instructor or Student). The proxy module 2324 may also allow pause and restart of a connection, session logs for assessment, and multiple connections to the remote devices 2312 (simultaneous access by student and instructor). As such, the proxy module may allow for the emulation of Instructor Led Training (ILT) training by allowing both an instructor and user to share the same Telnet session. This allows the instructor to "show" the user in realtime how to do

something on the device, or the instructor and other users can "watch" the user do something to the device. Also, multiple permissions may be set for multiple users. That is, some users may be permitted to both send and receive information to/from the device, while other users may just be permitted to receive information from the device. The proxy module 2324 further may permit pausing of the user's session in order to allow the operations module 2322 to perform operations on the device (e.g. scrub, load, or save). Thus, the user will not need to reconnect to the device every time such operations are performed.

Further, the proxy module may log information reflecting all or certain parts of a user's session. That is, storing the information that is sent to or from a device 2318. This session log can then be retrieved by the user at a later time. Also, the proxy module also preferably provides the capability to intercept and redirect or prevent potentially damaging user commands from reaching the remote devices 2318. For example, for routers the "write flash erase" can be potentially damaging to the router and as such, the proxy module can discard this command in order to prevent it from harming the device.

These software modules preferably communicate using an RPC, such as JavaTM RMI. RMI is used to receive messages from external systems such as the server and controller 2314 as well as being used for module to module communication. Further, as will be obvious to one of skill in the art, any other type of remote command method may be used in place of RMI. For example, CORBA may be used.

As discussed above, the RCM 2310 may use a database 2342 for storing overhead and status information, such as Sybase's Adaptive Server AnywhereTM (ASA) database. In a particular embodiment, this database 2342 contains the following tables:

Device_Type - a table that contains identification (ids) that identify device types (e.g. 1900,2500, Wintel PC)

Pod_Type - a table that contains ids that identify pod types.

Pod - a table that identifies the pods connected to this RCM.

Device - a table that contains detailed information about a specific device.

Device_User- a table that contains access and status information about a user.

Support_Device_Type - table that contains ids that identify support device types (e.g. a power controller, a console router, or a disk image server)

Support_Device - a table that contains detailed information about a specific support device

Figure 24 illustrates a logical block diagram of the proxy module 2324 of RCM 2310, in accordance with methods and systems consistent with the invention. As illustrated, the proxy module 2324 is connected to two user computers 2314 and an instructor computer 2315 over an Internet or Intranet 2332. As will be obvious to one of skill in the art, this connection may also be a direct connection, or some other type of connection. The proxy module is further connected to a database for storing session logs. Also connected to the proxy module 2324 is a device 2318. Preferably, the proxy module 2324 contains a device thread 2402. Further, the proxy module

preferably includes a buffer 2404, a write thread 2406, and a read thread 2408 for each user or instructor.

Although Figure 24 illustrates that the proxy module is connected to a single device 2318, in other embodiments it may be connected to multiple devices. In such embodiments, there is a separate device thread 2402 for each device 2318. Further, each user or instructor has a separate buffer 2404, write thread 2406, and read thread 2408 for each device connected to the proxy module 2324.

When a user wishes to communicate with the device 2318, it sends instructions to the respective read thread 2408 in the proxy module. The user may send this information using a particular application that depends on the specifics of the device 2318. For example, this information can be sent to the read thread by a user using Telnet, a web browser, or any other network application that the device 2318 requires for communications. The read thread then examines the instructions and depending on the users permissions either forwards the instruction to the device 2318 or discards it. Thus, potentially damaging user commands can be prevented from reaching the 2318. Further, different permissions may be set for different users and/or instructors. For example, it may be desirable that a certain user or instructor not be allowed to send instructions to the device 2318. Thus, the read thread 2408 for such a user can discard all instructions it receives from that user for the device 2318.

The device 2318 then receives the command 2318. This command will be echoed back to the device thread 2402 of the proxy module 2314. Further, all other device output is sent from the device 2318 to the device thread 2402 for the device.

These commands are then sent to the buffers 2404 for each user/instructor in communication with the device 2318. Depending on the permissions for the user computer 2314 or instructor computer 2315, the buffer may either send this information to a database 2346 that stores session logs, discard the information, or pass the information to the write thread 2406, which forward the information to its respective user computer 2314 or instructor computer 2318.

Thus, permissions may be set for each user computer 2314 and instructor computer 2315 that communicates with a particular device 2318, such that one user computer 2314 may be able to send and receive instructions and information to/from a device 2318, while other user computers 2314 are not permitted to send or receive information and/or instructions to/from the device 2318. This further permits an instructor terminal 2315 to watch how a particular user is exercising control over the device 2318, and then show a user(s) how they should control the device 2318.

While it has been illustrated and described what is at present considered to be the preferred embodiment and methods of the present invention, it will be understood by those skilled in the art that various changes and modifications may be made, and equivalents may be substituted for elements thereof without departing from the true scope of the invention.

In addition, many modifications may be made to adapt a particular element, technique or, implementation to the teachings of the present invention without departing from the central scope of the invention. Therefore, it is intended that this invention not be limited to the particular embodiment and methods disclosed herein, but that the invention includes all embodiments falling within the scope of the appended claims.

WHAT IS CLAIMED

1. A method for training users to operate a set of devices, wherein at least one client computer is connected to a device controller via a network and wherein a particular user uses the client computer to communicate with the device controller via the network to perform a training exercise using the set of one or more devices, the method comprising the steps of:

receiving control information at a proxy in the device controller reflecting at least one instruction from the client computer regarding a task to be performed as part of the training exercise;

performing an operation associated with at least one of the devices of the set of devices; and

providing a result of the operation to the client computer.

2. A system for training users to operate a set of one or more devices, comprising:
a device controller connected to at least one client computer via a network, wherein the device controller includes a proxy that receives control information from the client computer reflecting at least one task to be performed as part of a training exercise and transfers the control information to any device as part of the training exercise; and
a set devices that receives the control information transferred from the proxy in the device controller, performs an operation associated with the control information, and provides a result of the operation to the client computer via the device controller.

3. A method for training users to operate a set of devices, wherein at least one client computer is connected to a device controller via a network and wherein a particular user uses the client computer to communicate with the device controller via the network to perform a training exercise using the set of devices, the method comprising the steps of:

a server controller transferring overhead information using a generic set of commands to the device controller in order to perform overhead functions relating to the set of devices;

the device controller using the overhead information to exercise control over at least one of the devices in the set of devices for the purposes of performing overhead functions relating to the device;

receiving control information at the device controller reflecting at least one instruction from the client computer regarding a task to be performed as part of the training exercise;

performing an operation associated with at least one of the devices of the set of devices; and

providing a result of the operation to the client computer.

4. A system for training users to operate devices, comprising:

a set of devices;

a server/controller including a processor that transfers overhead information for overhead functions relating to the set of devices using a generic set of commands;

a device controller connected to at least one client computer via a network and connected to the server/controller, wherein the device controller includes a processor configured to receive

the overhead information from the server/controller, exercise control over the set of devices using the overhead information, receive from the client computer control information reflecting at least one task associated with a training exercise, and transfer the control information to the set devices in order to perform an operation associated with at least one device of the set.

ABSTRACT

Methods and Systems for training a user regarding controlling a device using a Resource Control Module. Through these methods and systems a user may remotely access a device in order to perform an assignment or task designed to train the user in the operation of the device. The user is able to access the devices through a user computer that is remotely connected to a Resource Control Module that may include a proxy. The Resource Control Module may also include a module for accepting overhead commands using a generic set of instructions from an external system.

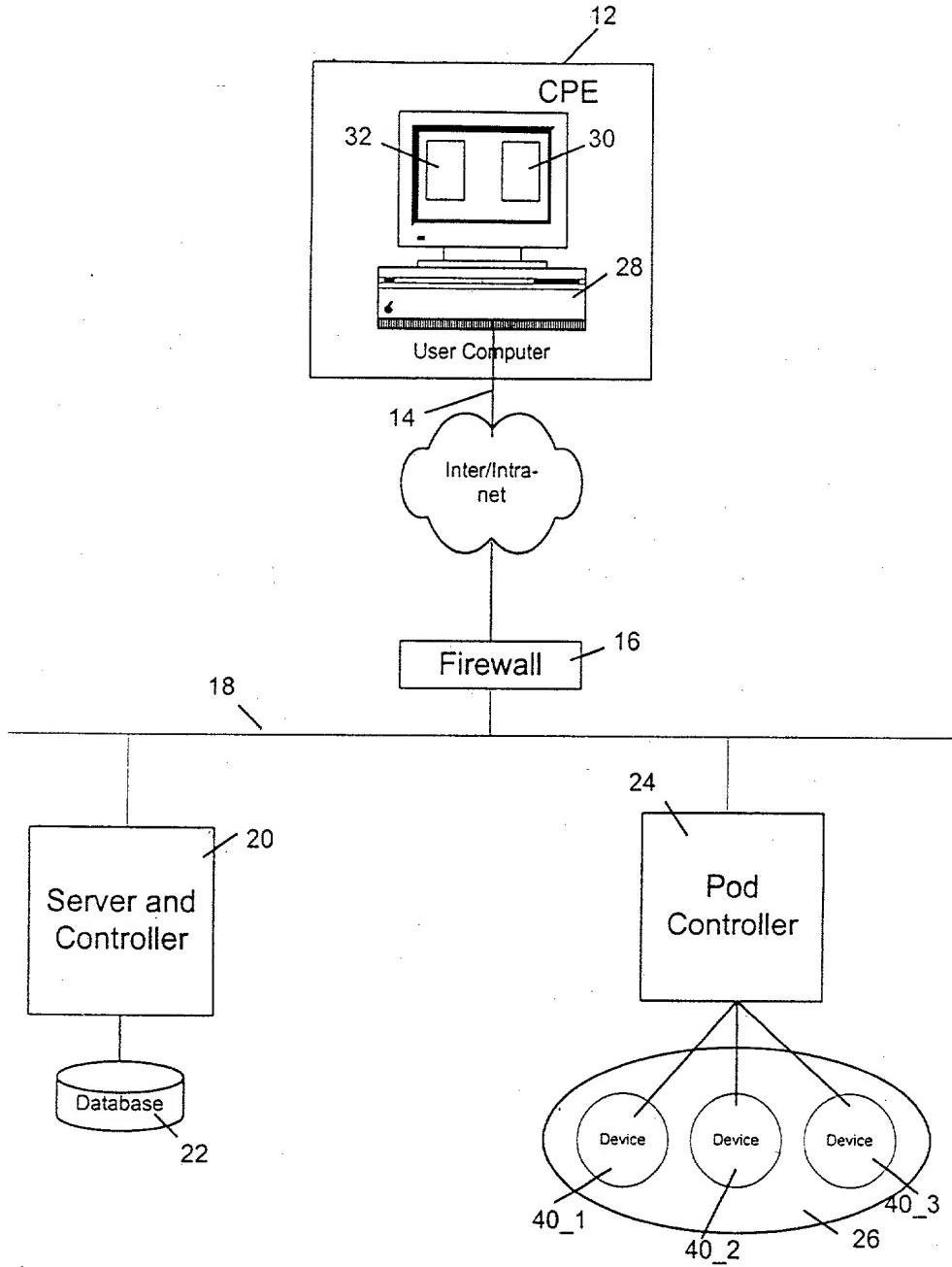


Figure 1

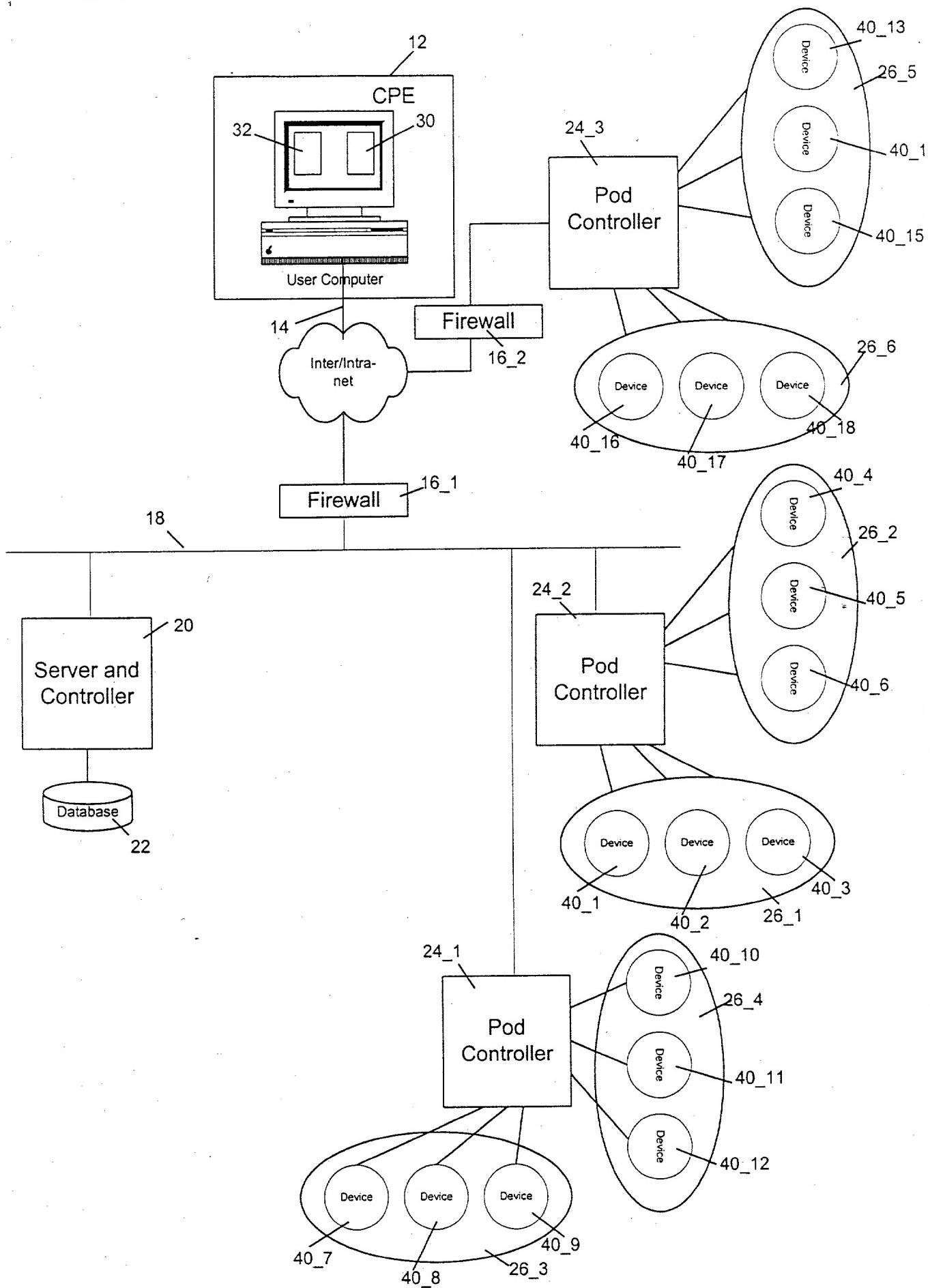


Figure 2

Pod Control System

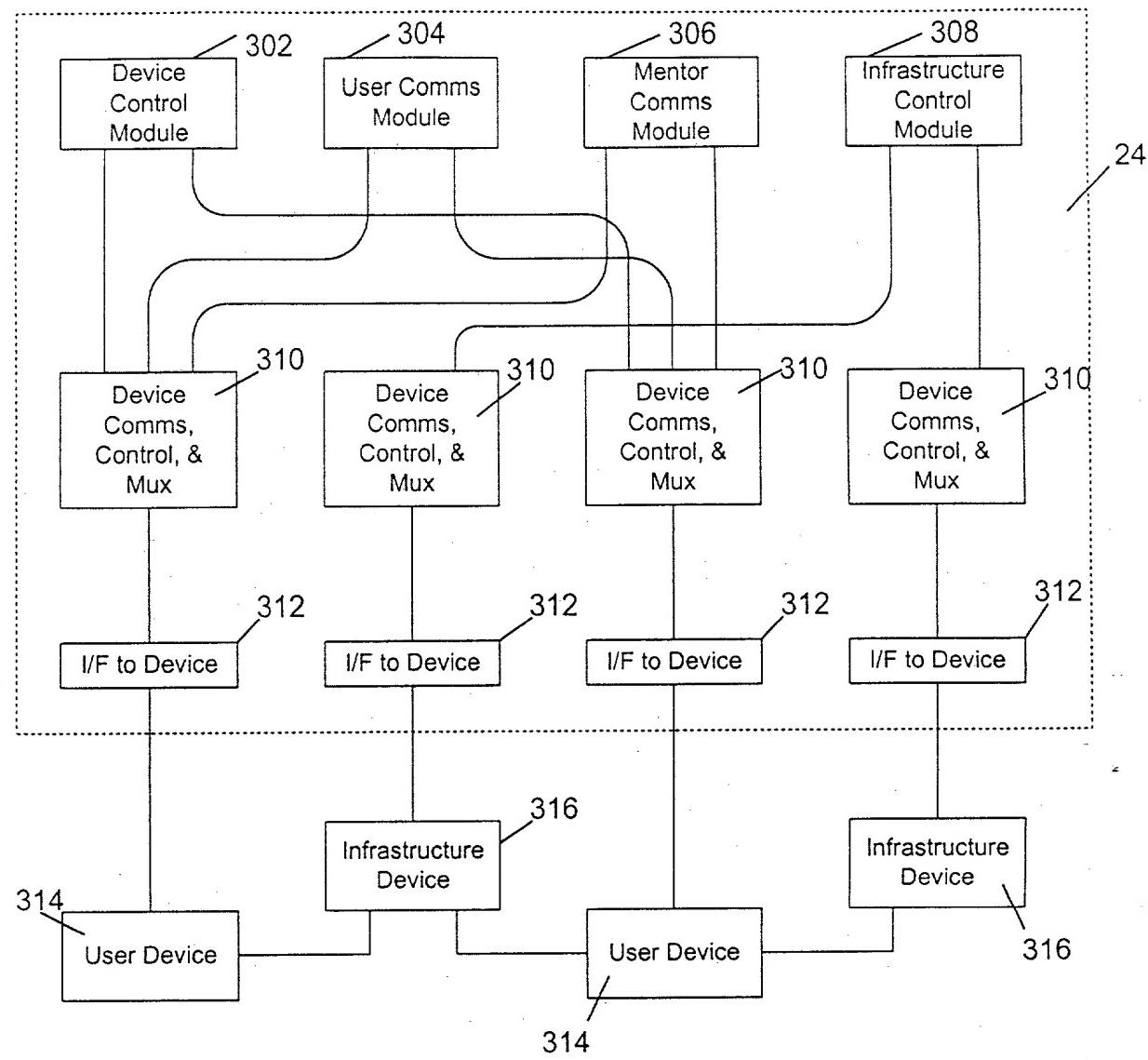


Figure 3

1011. Appletalk Routing

[Logout](#)

[Return to Locker](#)

Flash Controls **Reset** **Save** **Load** **Cycle Break** **Exit Console**

Reset | **Save** | **Load** | **Cycle Break**

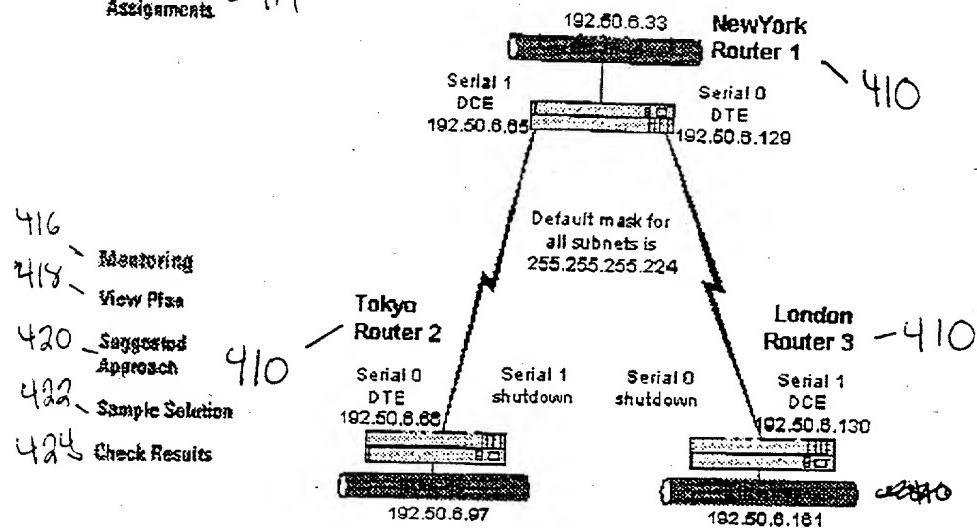
ମୁଖ୍ୟ ପ୍ରସାଦ

Time Remaining:

ଶ୍ରୀମତୀ କେନ୍ଦ୍ରପାତ୍ରାମାର୍ଥି

Instructions — 412

Scenarios & Assignments - 414



Click on a device above to open a console session.

Figure 4

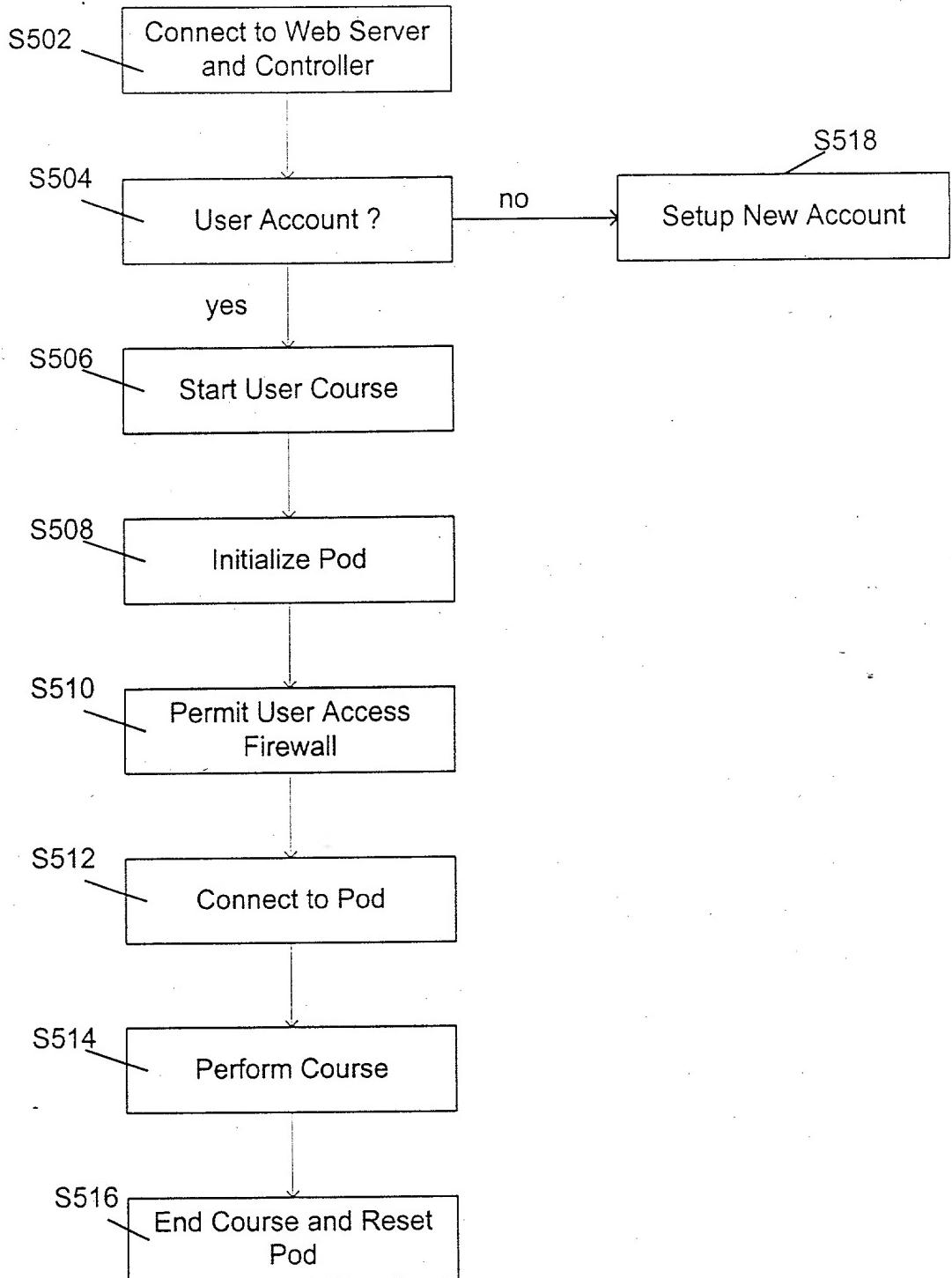


Figure 5

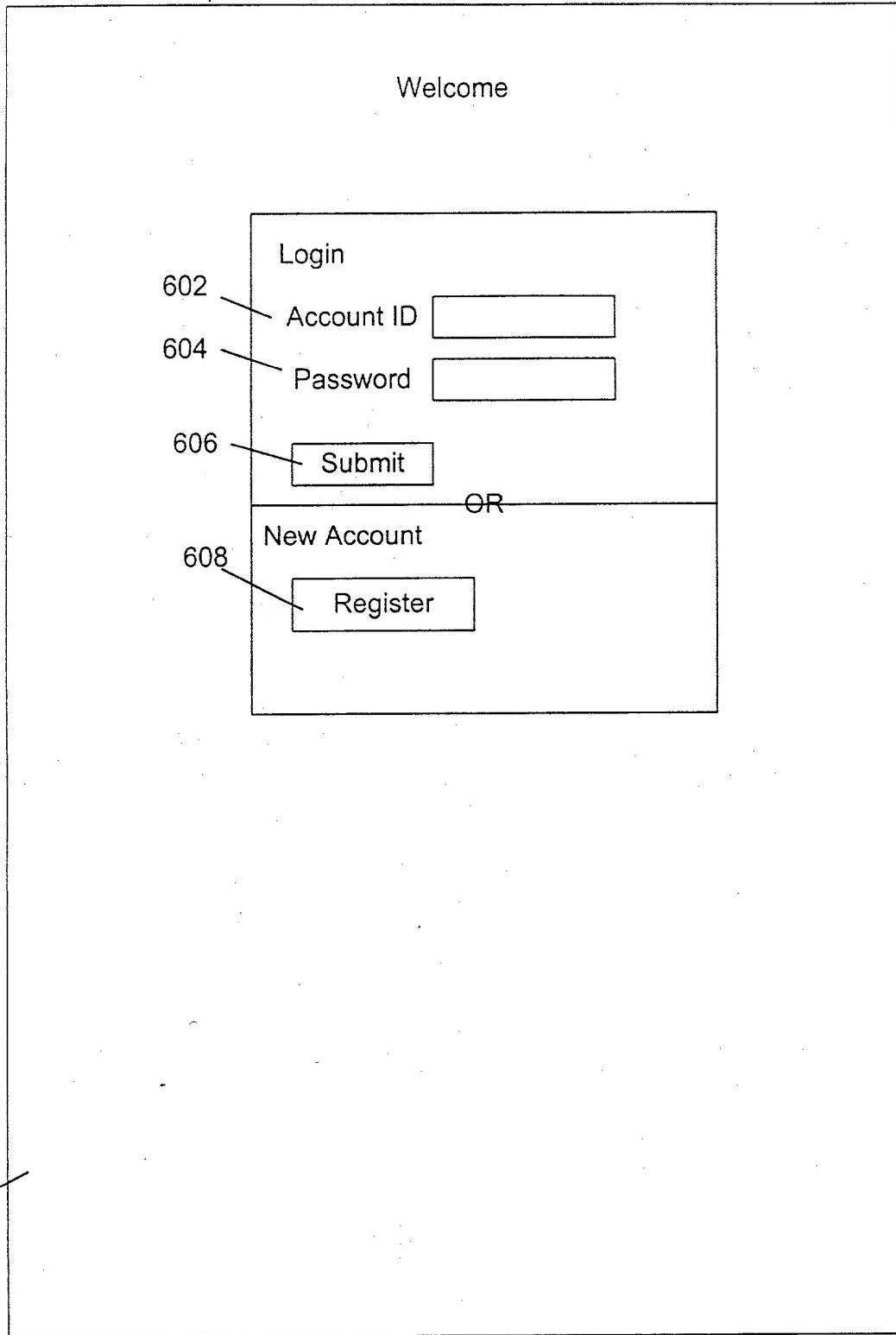


Figure 6

| | | |
|----------------------|----------------------|----------------------|
| 1. Name / 702 | | |
| Last Name | First Name | M.I. |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| 2. Address / 704 | | |
| Company | <input type="text"/> | |
| Street | <input type="text"/> | |
| City | <input type="text"/> | |
| State | <input type="text"/> | |
| 3. User Name / 706 | | |
| <input type="text"/> | | |
| 4. Password / 708 | | |
| <input type="text"/> | | |

Figure 7

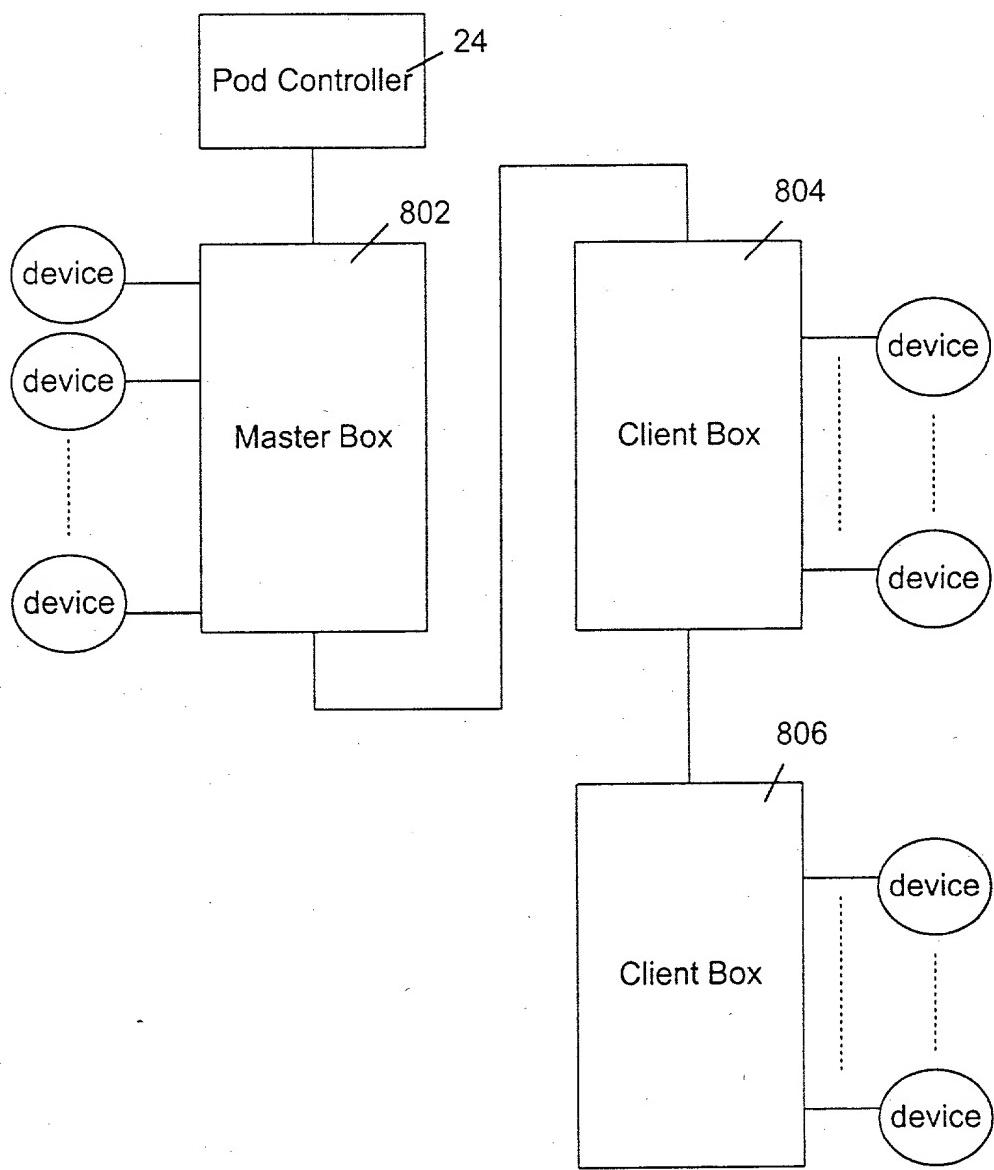


Figure 8

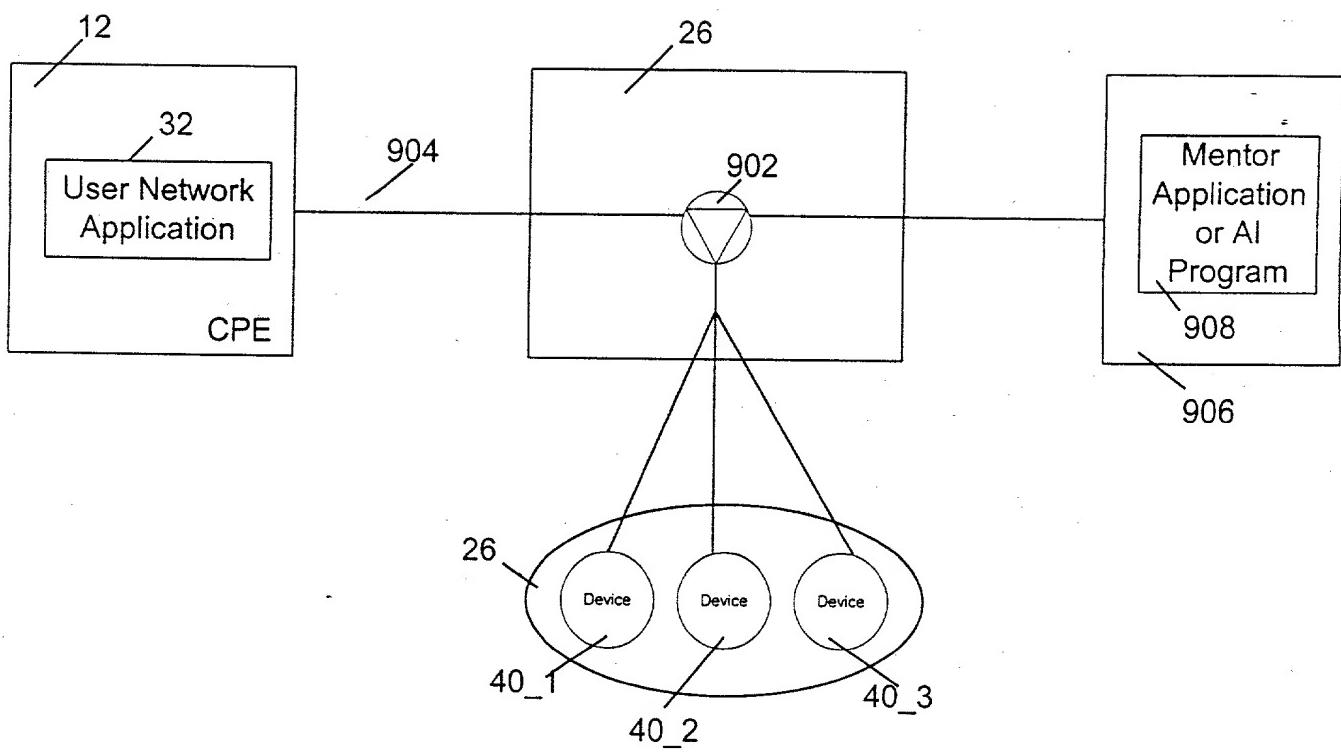


Figure 9

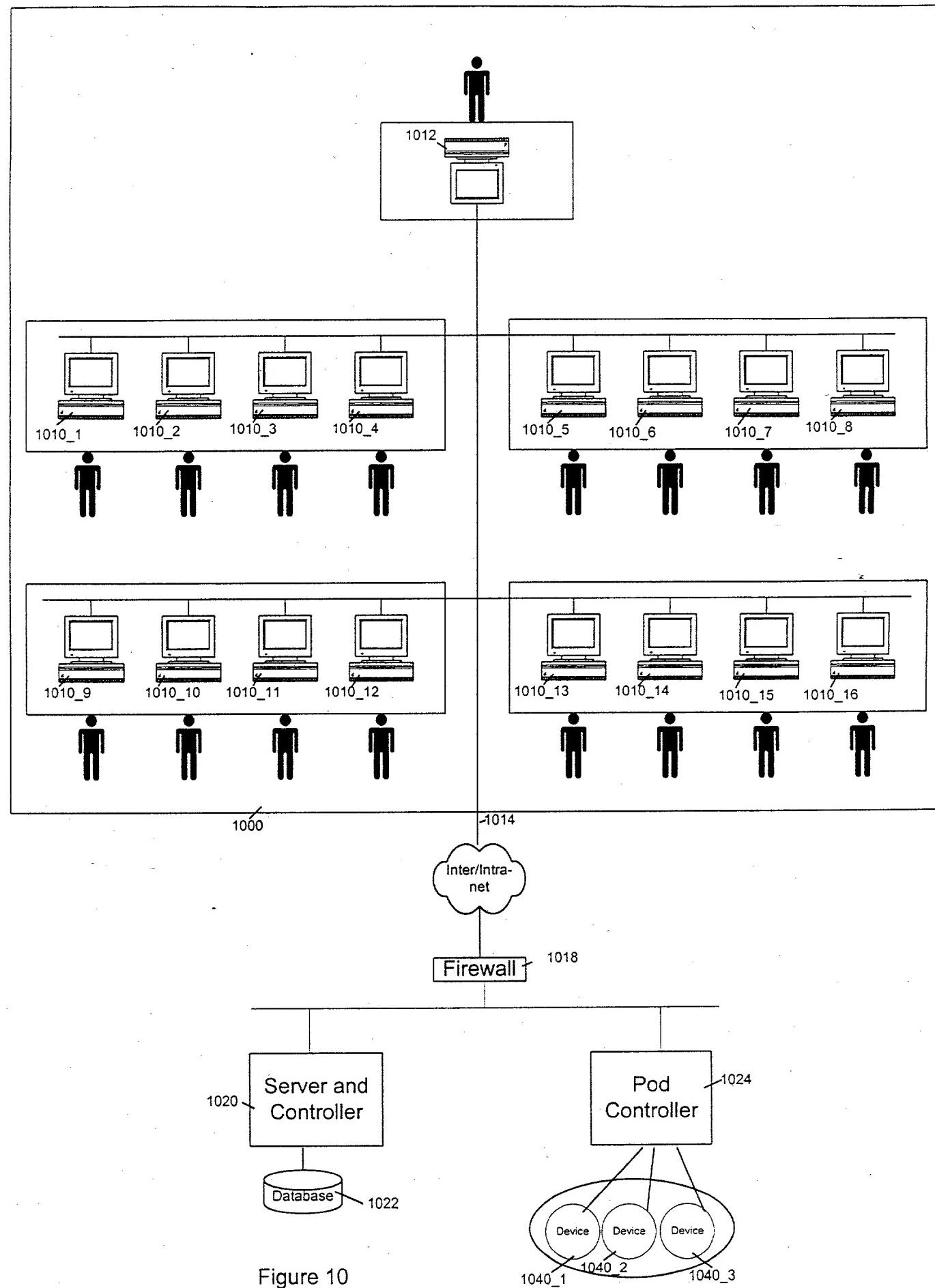


Figure 10

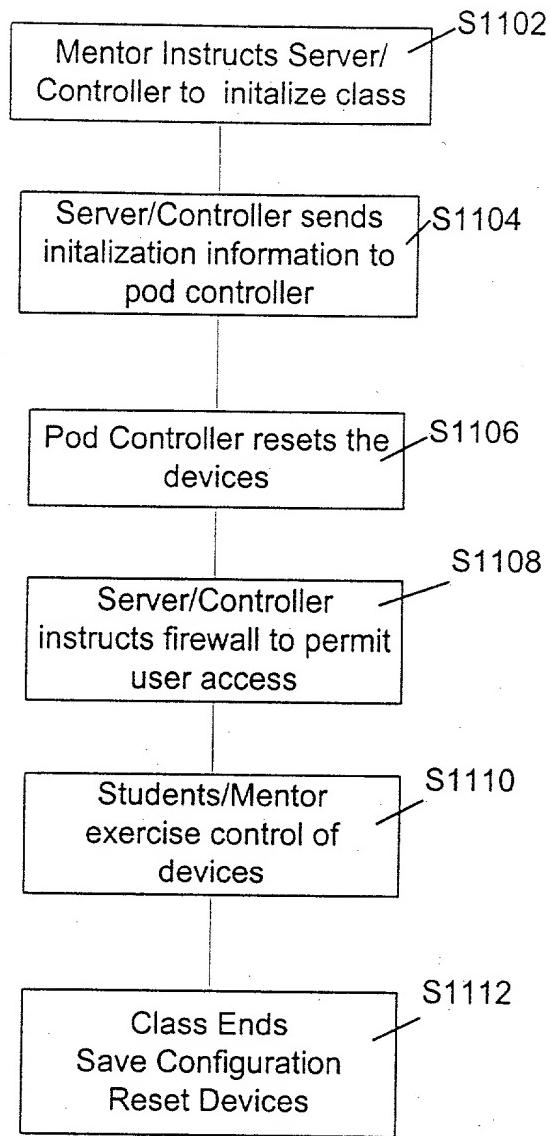


Figure 11

Learning Structure

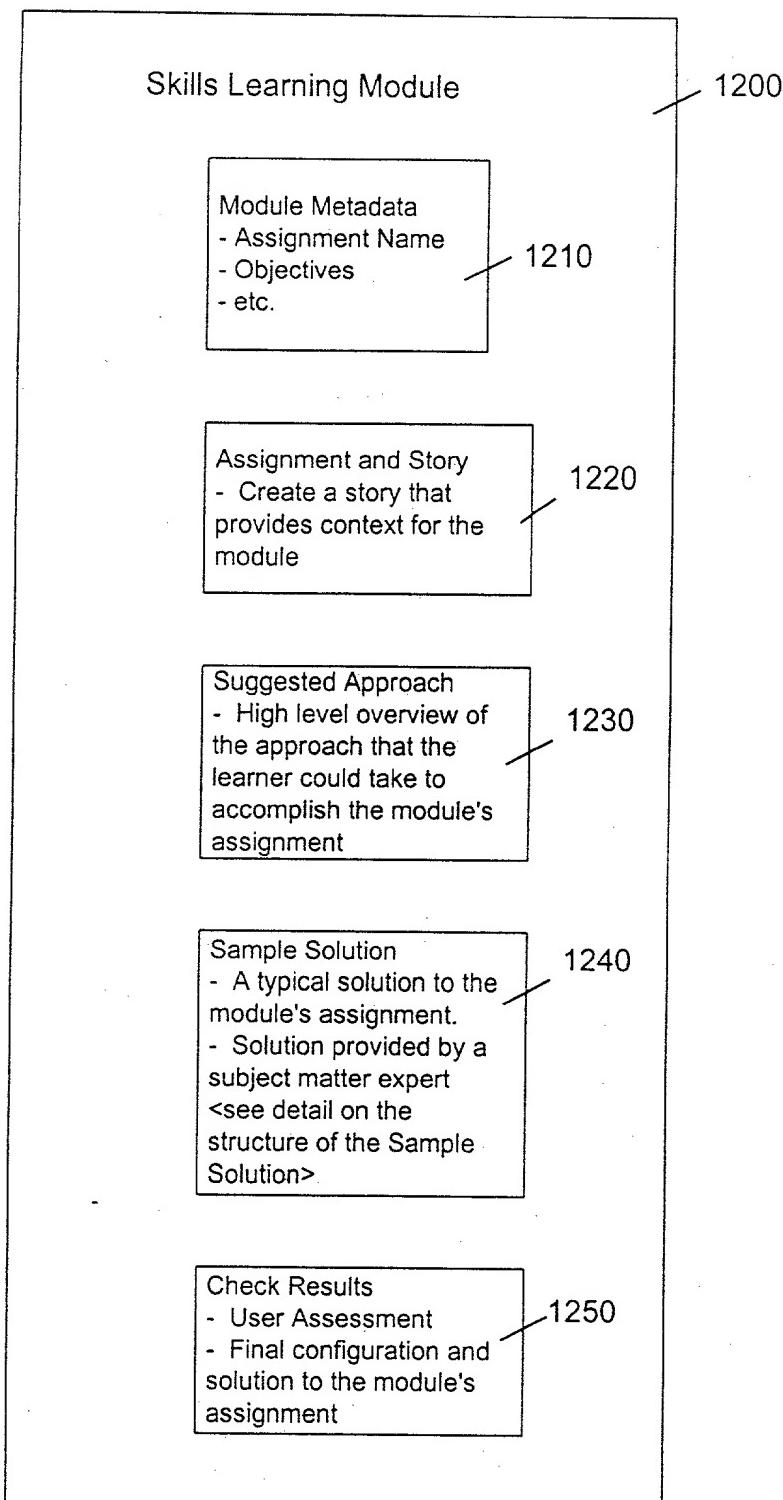


Figure 12

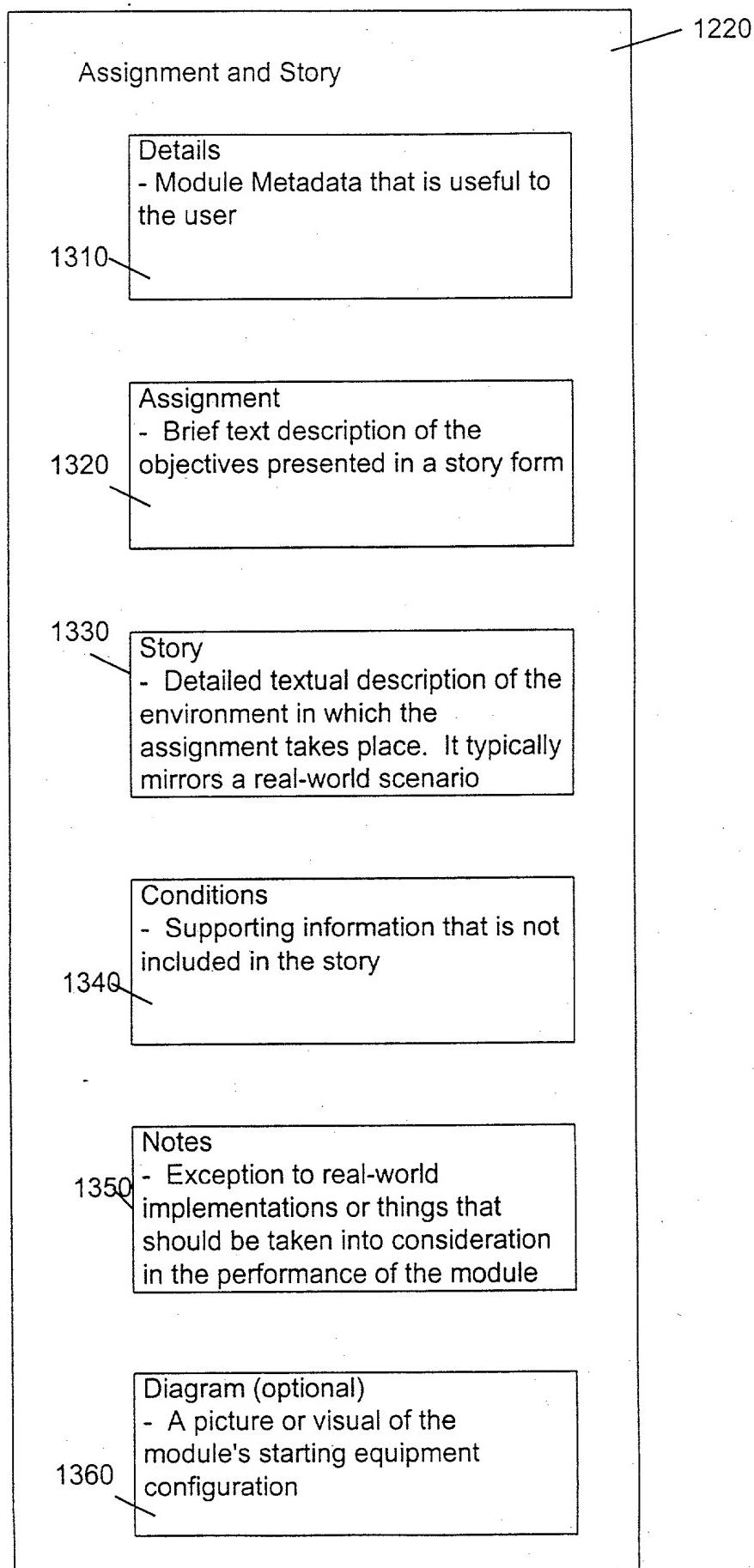


Figure 13

scenario | Lab Highlights | Story | Conditions | Diagram

1011. Appletalk Routing

1410
Details

| | |
|--------------------------|---|
| vLab Title | 1011. Appletalk Routing |
| Technology | Network Layer |
| Level of Difficulty | Basic |
| Time Required | 57 mins |
| Certification | CCNA |
| Desired Learner Outcomes | Experience designing and implementing Appletalk in a network. |
| Desired Network Outcomes | Appletalk routing is operational on the network. |

Top

1420

Assignment

Design an Appletalk numbering plan and enable Appletalk routing

Top

Story

Your network manager has told you that your network will soon have to carry Appletalk traffic. In order for this to happen you must plan an Appletalk numbering scheme and assign Appletalk zone names for each of the segments in your network. You will also enable Appletalk routing on all of the active interfaces on your routers. Once Appletalk is enabled on the routers and configured on the interfaces, you should verify that Appletalk is functioning properly.

Top

1430

Conditions

IP routing is already up and running on this network. DO NOT CHANGE ANY OF THE IP ROUTING CONFIGURATIONS.

Your Apple administrator has given you the following range of Appletalk network numbers, 2000 – 2999. You may use any number within that range to assign a unique Appletalk network number to each segment in the network. All of the serial links should be configured in the 'cereal zone'. You should make up unique zone names for each of the Ethernet interfaces.

Top

1440

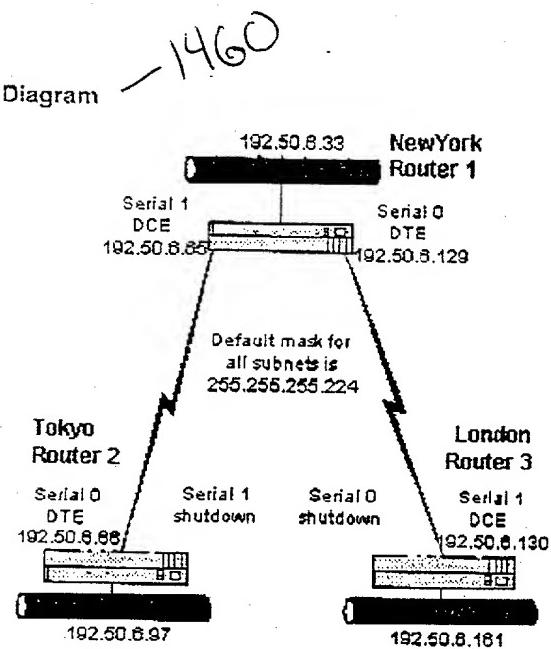
Notes

The serial links between routers are implemented via direct connections in this lab and do not actually connect through any leased line services for the serial links. Here is the existing IP network. Use this as a starting point to plan your Appletalk Network.

Top

Figure 14-a

Diagram



Top

Figure 14-b

Suggested Approach

1011. Appletalk Routing

Figure out the Appletalk numbering plan. Assign a unique Appletalk cable range to each network segment. Note the Appletalk zone names on each network. Enable Appletalk routing on the routers, then configure the appropriate Appletalk cable range on each active router interface. Once that is done verify proper Appletalk operation using show commands.

Figure 15

Sample Solution

1602 Task 1 - <task 1 description>
1602 Task 2 - <task 2 description>
1602 Hint
1604 Step 1 - <step 1 description>
1606 Action - <What the user should do to accomplish this step>
Result - <What result should be expected from performing
the action>
1606 Explanation - <Why the action was needed>
Step 2 - <step 2 description>
: : : :
: : : :
: : : :
1602 Task 3 - <task 3 description>
: : : :
: : : :
: : : :

Figure 16

sample solution

1602 — Plan Appletalk addressing

1604 — **HINT**

1606 — Assign on paper a unique Appletalk network number to each network segment

Action: Choose a cable range from the addresses that were given to you by the Appletalk administrator (2000 – 2999) for each network segment.

Result: Each 'wire' in the network should get a different Appletalk cable range.

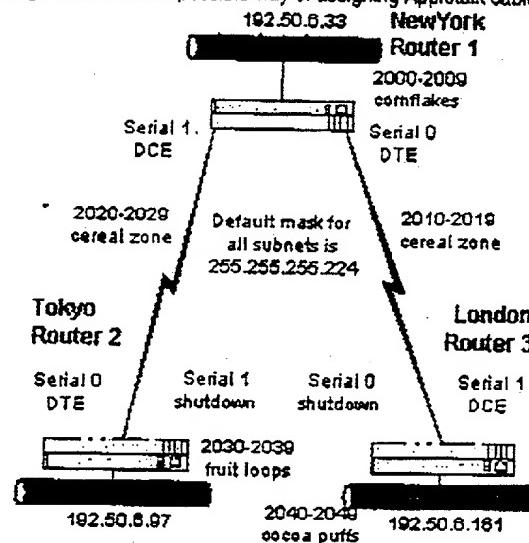
Explanation: Appletalk routing requires that every segment (or wire) in the network have a unique cable range in order for the Appletalk protocol to identify each part (link) of the network. A cable range is a contiguous range of network numbers that is assigned to a network segment. An example of a cable range would be 2300 – 2310. This assigns the range of network numbers from 2300 to 2310 to the network segment. Once you have a completed diagram, note the interfaces that each link connects to.

1606 — Assign on paper Appletalk zone names to each network segment, and assign all of the serial links in the 'cereal zone'.

Action: You need to think up three additional unique zone names for each of the Ethernet segments.

Result: An Appletalk zone can cover more than one network segment. Each network segment must be in at least one Appletalk zone. Zones are alpha numeric names, spaces are legal characters. Router ports that connect to the same network segment must be configured identically.

Explanation: The sample diagram shows one possible way of assigning Appletalk cable ranges and zone names to the various links in



1602 — Enable Appletalk Routing on each router

1604 — **HINT**

1606 — Start the Appletalk routing processes on the New York router.

Action: appletalk routing

Result:

```
NewYork>en
NewYork#conf t
Enter configuration commands, one per line. End with CNTL/Z.
NewYork(config)#appletalk routing
NewYork(config)#^Z
NewYork#
SYS-5-CONFIG_I: Configured from console by console
```

Explanation: The Appletalk routing process is not on by default. You must tell the router that you want it to route Appletalk packets. The "Appletalk routing" command also starts the Appletalk RTMP routing protocol running.

1606 — Start the Appletalk routing processes on the Tokyo router.

Figure 17-a

Action: appletalk routing
Result:
Tokyo>en
Tokyo#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Tokyo(config)#appletalk routing
Tokyo(config)#^Z
Tokyo#
%SYS-5-CONFIG_I: Configured from console by console

Explanation: The Appletalk routing process is not on by default. You must tell the router that you want it to route Appletalk packets. The "Appletalk routing" command also starts the Appletalk RTMP routing protocol running.

1606 — Start the Appletalk routing processes on the London router.

Action: appletalk routing
Result:
London>en
London#conf t
Enter configuration commands, one per line. End with CNTL/Z.
London(config)#appletalk routing
London(config)#^Z
London#
%SYS-5-CONFIG_I: Configured from console by console

Explanation: The Appletalk routing process is not on by default. You must tell the router that you want it to route Appletalk packets. The "Appletalk routing" command also starts the Appletalk RTMP routing protocol running.

1606 — Configure the proper Appletalk network number on each interface (Ethernet 0, Serial 0, and/or Serial 1) for the New York router using the diagram you made earlier.

Action:
appletalk cable-range <cable range>
appletalk zone <zone name>

Result:
NewYork#
%SYS-5-CONFIG_I: Configured from console by console
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
NewYork(config)#interface serial 0
NewYork(config-if)#appletalk cable-range 2010-2019
NewYork(config-if)#appletalk zone cereal zone
NewYork(config-if)#interface serial 1
NewYork(config-if)#appletalk cable-range 2020-2029
NewYork(config-if)#appletalk zone cereal zone
NewYork(config-if)#interface ethernet 0
NewYork(config-if)#appletalk cable-range 2000-2009
NewYork(config-if)#appletalk zone cornflakes
NewYork(config-if)#^Z
NewYork#
%SYS-5-CONFIG_I: Configured from console by console

Explanation: A unique Appletalk cable range must be assigned to each interface routing packets for the Appletalk protocol.

1606 — Configure the proper Appletalk network number on each Interface (Ethernet 0, Serial 0, and/or Serial 1) for the Tokyo router using the diagram you made earlier.

Figure 17-6

Action:

```
appletalk cable-range <cable range>
appletalk zone <zone name>
```

Result:

```
Tokyo#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Tokyo(config)#int e 0
Tokyo(config-if)#appletalk cable-range 2030-2039
Tokyo(config-if)#appletalk zone fruit loops
Tokyo(config-if)#int s 0
Tokyo(config-if)#appletalk cable-range 2020-2029
Tokyo(config-if)#appletalk zone cereal zone
Tokyo(config-if)#^Z
Tokyo#
%SYS-5-CONFIG_I: Configured from console by console
```

Explanation: A unique Appletalk cable range must be assigned to each interface routing packets for the Appletalk protocol.

- 1606 — Configure the proper Appletalk network number on each interface (Ethernet 0, Serial 0, and/or Serial 1) for the London router using the diagram you made earlier.

Action:

```
appletalk cable-range <cable range>
appletalk zone <zone name>
```

Result:

```
London#conf t
Enter configuration commands, one per line. End with CNTL/Z.
London(config)#int e 0
London(config-if)#appletalk cable-range 2040-2049
London(config-if)#appletalk zone cocoa puffs
London(config-if)#int s 1
London(config-if)#appletalk cable-range 2010-2019
London(config-if)#appletalk zone cereal zone
London(config-if)#^Z
London#
%SYS-5-CONFIG_I: Configured from console by console
```

Explanation: A unique Appletalk cable range must be assigned to each interface routing packets for the Appletalk protocol.

- 1602 — Verify Proper Operation of Appletalk Routing

HINT

- 1604 — Use a brief version of a show command to see that the Appletalk protocol is properly configured and running on the New York router.

Action: show appletalk interface brief

Result:

| Interface | Address | Config | Status/Line Protocol | Atalk Protocol |
|-----------|------------|--------------|-----------------------|----------------|
| BRI0 | unassigned | not config'd | administratively down | n/a |
| BRI0:1 | unassigned | not config'd | administratively down | n/a |
| BRI0:2 | unassigned | not config'd | administratively down | n/a |
| Ethernet0 | 2002.14 | Extended | up | up |
| Serial0 | 2010.174 | Extended | up | up |
| Serial1 | 2025.55 | Extended | up | up |
| Serial2 | unassigned | not config'd | administratively down | n/a |
| Serial3 | unassigned | not config'd | administratively down | n/a |

Explanation: The three interfaces you configured (E0, S0 and S1) on router 1 (NewYork) all show that they are 'up'. This means that they are properly configured and operational. This is a good quick check to see if the Appletalk protocol is running. If one of the interfaces that you have configured is 'down', check to be sure that the interface at the other end of the link has the same Appletalk cable range configured on it. The number after the cable-range number is the host number. The host number is dynamically assigned and will probably be different in your display.

- 1606 — Use a brief version of a show command to see that the Appletalk protocol is properly configured and running on the Tokyo router.

Figure 17-c

Action: show appletalk interface brief

Result:

```
Tokyo#sh appletalk interface brief
Interface      Address    Config      Status/Line Protocol Atalk Protocol
BRI0          unassigned  not config'd  administratively down n/a
BRI0:1        unassigned  not config'd  administratively down n/a
BRI0:2        unassigned  not config'd  administratively down n/a
Ethernet0     2038.37   Extended    up           up
Serial0        2022.76   Extended    up           up
Serial1        unassigned  not config'd  administratively down n/a
Serial2        unassigned  not config'd  administratively down n/a
Serial3        unassigned  not config'd  administratively down n/a
```

Explanation: The two interfaces you configured (E0, and S0) on router 2 (Tokyo) all show that they are 'up'. This means that they are properly configured and operational. This is a good quick check to see if the Appletalk protocol is running. If one of the interfaces that you have configured is 'down', check to be sure that the interface at the other end of the link has the same Appletalk cable range configured on it. The number after the cable-range number is the host number. The host number is dynamically assigned and will probably be different in your display.

1606 - Use a brief version of a show command to see that the Appletalk protocol is properly configured and running on the London router.

Action: show appletalk interface brief

Result:

```
London#show appletalk interface brief
Interface      Address    Config      Status/Line Protocol Atalk Protocol
BRI0          unassigned  not config'd  administratively down n/a
BRI0:1        unassigned  not config'd  administratively down n/a
BRI0:2        unassigned  not config'd  administratively down n/a
Ethernet0     2045.215   Extended   up           up
Serial0        unassigned  not config'd  administratively down n/a
Serial1        2013.235   Extended   up           up
Serial2        unassigned  not config'd  administratively down n/a
Serial3        unassigned  not config'd  administratively down n/a
```

Explanation: The two interfaces you configured (E0, and S1) on router 3 (London) all show that they are 'up'. This means that they are properly configured and operational. This is a good quick check to see if the Appletalk protocol is running. If one of the interfaces that you have configured is 'down', check to be sure that the interface at the other end of the link has the same Appletalk cable range configured on it. The number after the cable-range number is the host number. The host number is dynamically assigned and will probably be different in your display.

1606 - Use a show Appletalk command to view all of the Appletalk parameters of a particular interface.

Action: show Appletalk interface

Result:

```
NewYork#show appletalk interface serial 0
Serial0 is up, line protocol is up
  AppleTalk cable range is 2010-2019
  AppleTalk address is 2010.174, Valid
  AppleTalk zone is "cereal zone"
  AppleTalk port configuration verified by 2013.235
  AppleTalk address gleaning is not supported by hardware
  AppleTalk route cache is enabled
```

Explanation: The important thing to note here is that the interface shows 'up' and line protocol is 'up'. This means the interface is communicating with the network it is connected to. You can also see the Appletalk address of this interface on the fourth line of the example. You can also see that the configuration of this port has been verified by the router at the other end of the link.

1606 - Use the 'show Appletalk route' command to look at the Appletalk routing table

Figure 17-d

Action: show appletalk route

Result:

```
NewYork#show appletalk route
Codes: R - RTMP derived, E - EIGRP derived, C - connected, A - AURP
      S - static P - proxy
5 routes in internet
```

The first zone listed for each entry is its default (primary) zone.

```
C Net 2000-2009 directly connected, Ethernet0, zone cornflakes
C Net 2010-2019 directly connected, Serial0, zone cereal zone
C Net 2020-2029 directly connected, Serial1, zone cereal zone
R Net 2030-2039 [1/G] via 2022.76, 2 sec, Serial1, zone fruit loops
R Net 2040-2049 [1/G] via 2013.235, 0 sec, Serial0, zone cocoa puffs
NewYork#
```

Explanation: After the routing updates propagate (roughly 90 seconds), each router should have five Appletalk routes in its routing table. If they do not, make sure that the routers are properly configured.

Copyright © 1998, 1999 MentorLabs, LLC.

Figure 17-2

Figure 18-a

check results

Router 1

```
hostname Washington
!
enable password cisco
!
interface Ethernet0
ip address 10.23.0.1 255.255.0.0
no keepalive
no shutdown
!
interface Serial0
ip address 10.33.0.2 255.255.0.0
ip mroute-cache
no shutdown
!
interface Serial1
ip address 10.29.0.1 255.255.0.0
clockrate 56000
no shutdown
!
interface Serial2
no ip address
shutdown
!
interface Serial3
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
router rip
network 10.0.0.0
!
no ip classless
!
banner motd %
```

IP RIP Foundation Lab Router1
Version: 1.0
Date: July 10, 1998
Copyright 1998, 1999, MentorLabs, LLC

All rights reserved %

Passwords:
User - cisco
Enable - cisco

```
!
line con 0
password cisco
login
line aux 0
line vty 0 4
password cisco
login
!
end
```

Router 2

Figure 18-b

```
hostname Minot
!
enable password sanfran
!
interface Ethernet0
 ip address 10.30.0.1 255.255.0.0
 no keepalive
 no shutdown
!
interface Serial0
 ip address 10.29.0.2 255.255.0.0
 ip route-cache
 no shutdown
!
interface Serial1
 ip address 10.31.0.1 255.255.0.0
 clockrate 56000
 no shutdown
!
interface Serial2
 no ip address
 shutdown
!
interface Serial3
 no ip address
 shutdown
!
Interface BRI0
 no ip address
 shutdown
!
router rip
 network 10.0.0.0
!
ip classless
!
!
banner motd %
!
```

IP RIP Foundation Lab Router2
Version: 1.0
Date: July 10, 1998

Copyright 1998, 1999, MentorLabs, LLC
All rights reserved %

Passwords:
User - cisco
Enable - sanfran
!
line con 0
password cisco
login
line aux 0
line vty 0 4
password cisco
login
!
end

Router 3

```
!
hostname Leesville
!
enable password sanfran
!
interface Ethernet0
 ip address 10.32.0.1 255.255.0.0
 no keepalive
 no shutdown
!
```

Figure 18-c

```
interface Serial0
ip address 10.31.0.2 255.255.0.0
ip mroute-cache
no shutdown
!
Interface Serial1
ip address 10.33.0.1 255.255.0.0
clockrate 56000
no shutdown
!
interface Serial2
no ip address
shutdown
!
interface Serial3
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
router rip
network 10.0.0.0
!
ip classless
!
!
banner motd %
```

IP RIP Foundation Lab Router3
Version: 1.0
Date: July 10, 1998
Copyright 1998, 1999, MentorLabs, LLC

All rights reserved %

Passwords:
User - cisco
Enable - sanfran
!
line con 0
password cisco
login
line aux 0
line vty 0 4
password cisco
login
!
end

Copyright © 1998, 1999 MentorLabs, LLC.

check results

Check your configuration to confirm the network is operating per the Story and Conditions.
(Use appropriate show, debug, and ping commands to verify network operations.)

HINT

Verify that the physical links in the network are running.

Action: show ip interface brief
Result: Hub#

| Interface | IP-Address | OK? | Method | Status | Protocol |
|-----------|---------------|-----|--------|-----------------------|----------|
| BRI0 | unassigned | YES | unset | administratively down | down |
| BRI0:1 | unassigned | YES | unset | administratively down | down |
| BRI0:2 | unassigned | YES | unset | administratively down | down |
| Ethernet0 | 192.168.2.129 | YES | unset | administratively down | down |
| Serial0 | 172.18.1.33 | YES | manual | up | up |
| Serial1 | 192.168.2.66 | YES | manual | up | up |
| Serial2 | unassigned | YES | manual | up | up |
| Serial3 | unassigned | YES | unset | administratively down | down |
| | | YES | unset | administratively down | down |

Hub#

Explanation: The configured interfaces should all have up for Status and up for Protocol. If not, use other show commands to determine why.
Confirm the routing table on Branch_1 supports the Story and Conditions.

Action: show ip route
Result: Branch_1#
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is 192.168.2.66 to network 172.18.0.0

I* 172.18.0.0/16 [100/82125] via 192.168.2.66, 00:00:11, Serial0
192.168.2.0/28 is subnetted, 3 subnets
C 192.168.2.64 is directly connected, Serial0
C 192.168.2.192 is directly connected, Ethernet0
I 192.168.2.128 [100/80225] via 192.168.2.66, 00:00:12, Serial0

Branch_1#

Explanation: Except for the time since last routing update, your routing table on Branch_1 should match the Results above. Do your metrics well?

Note that the Gateway of last resort and the candidate default route most both appear.

Confirm the routing table on the ISP supports the Story and Conditions.

Figure 19-a

Action: show ip route
Result: ISP#sh ip ro
Codes: C - Connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

C 172.18.0.0/30 is subnetted, 1 subnets
C 172.18.1.32 is directly connected, Serial1
10.0.0.0/24 is subnetted, 1 subnets
C 10.1.3.0 is directly connected, Ethernet0
S 192.168.2.0/24 [1/0] via 172.18.1.33
ISP#

Explanation: The ISP should have three subnets listed.

Confirm the routing table on the Hub supports the Story and Conditions:

Action: show ip route
Result: (There are two main possible results, depending on how the default route was config
Hub#sh ip ro

Codes: C - Connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is 172.18.1.34 to network 0.0.0.0

* 172.18.0.0/30 is subnetted, 1 subnets
C 172.18.1.32 is directly connected, Serial0
192.168.2.0/28 is subnetted, 3 subnets
C 192.168.2.64 is directly connected, Serial1
I 192.168.2.192 [100/80225] via 192.168.2.65, 00:00:13, Serial1
C 192.168.2.128 is directly connected, Ethernet0
S* 0.0.0.0/0 [1/0] via 172.18.1.34
Hub#

. . . or . . .

Hub#sh ip ro
Codes: C - Connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

* 172.18.0.0/30 is subnetted, 1 subnets
C 172.18.1.32 is directly connected, Serial0
192.168.2.0/28 is subnetted, 3 subnets
C 192.168.2.64 is directly connected, Serial1
I 192.168.2.192 [100/80225] via 192.168.2.65, 00:00:19, Serial1
C 192.168.2.128 is directly connected, Ethernet0
S* 0.0.0.0/0 is directly connected, Serial0
Hub#

Explanation: Both options for configuring a default route will support the network.

Note : Do your metrics match as well?

Verify that the network is operating as described in the Story and Conditions.

Figure 19-1

Action: ping ip-address
 Result: Branch_1#ping 10.1.3.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
```

Branch_1#

```
ISP#ping
Protocol [ip]:
Target IP address: 192.168.2.129
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.3.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/17/20 ms
```

```
ISP#ping
Protocol [ip]:
Target IP address: 192.168.2.193
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.3.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.193, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
```

ISP#

Explanation: Your ping tests from Branch_1 to the subnet 10.1.3.0 should be successful. Extended ping tests from the ISP's Ethernet to the Ethernet and Branch_1's Ethernet should also be successful.

Verify that the routing updates have been minimized as described in the Story and Conditions.

Action: debug ip packet
 Result: ISP#debug ip packet
 IP packet debugging is on
 ISP#

```
ISP#no debug ip packet
IP packet debugging is off
ISP#
```

Explanation: The debugging information should be quiet after several minutes. If so, you can turn off IP packet debugging, and know that IG routing packets are not being sent to the ISP.

Copyright © 1998, 1999 MentorLabs, LLC.

Figure 19-C

1002. Connectivity Between Routers
vLab Archive

▼ Archive History

▼ Archive Date

Date Lab Started : 1999-Jul-15 16:06:40.864802

Date Lab Completed : 1999-Jul-15 16:09:49.268665

Date Lab Archived : 1999-Jul-15 16:10:23.670189

▼ Lab Information

2002 — Plan

2004 — Debrief

2006 — Saved Configs

Figure 20

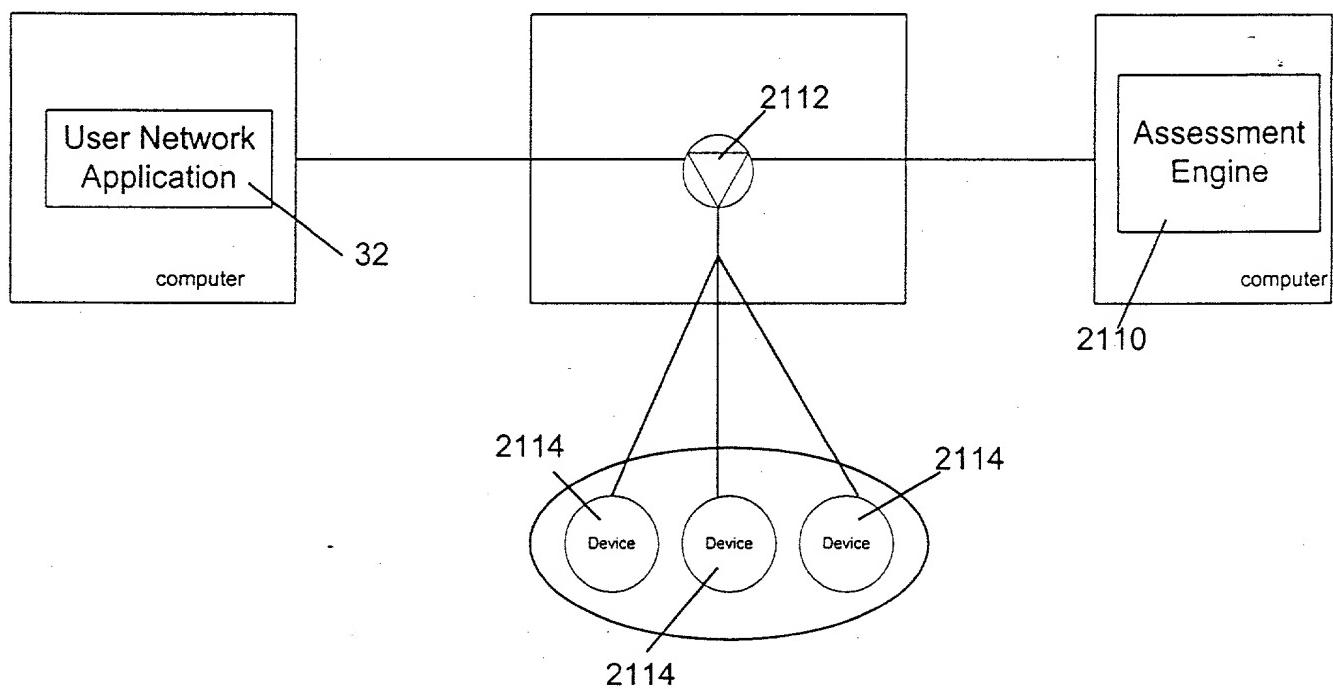


Figure 21

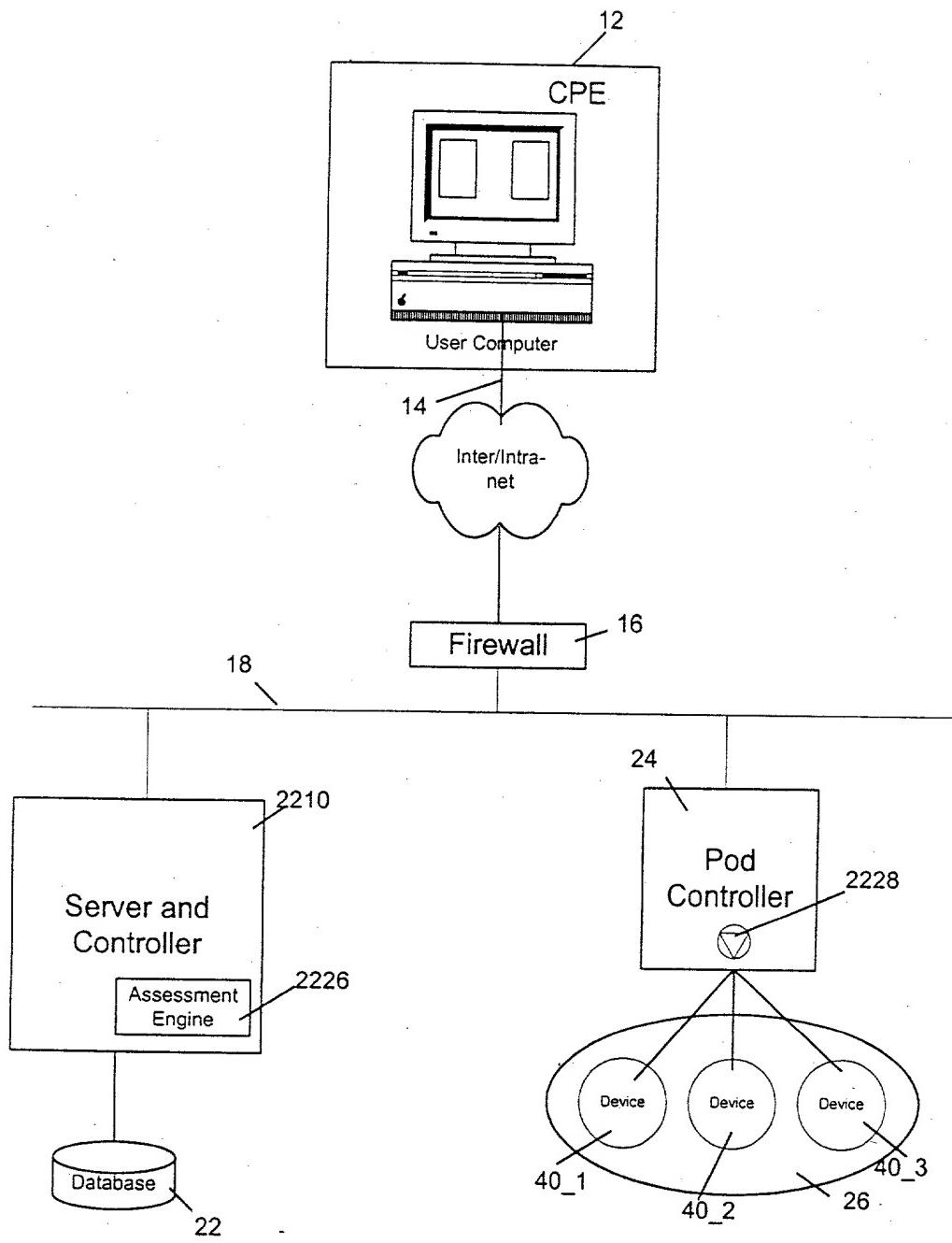


Figure 22

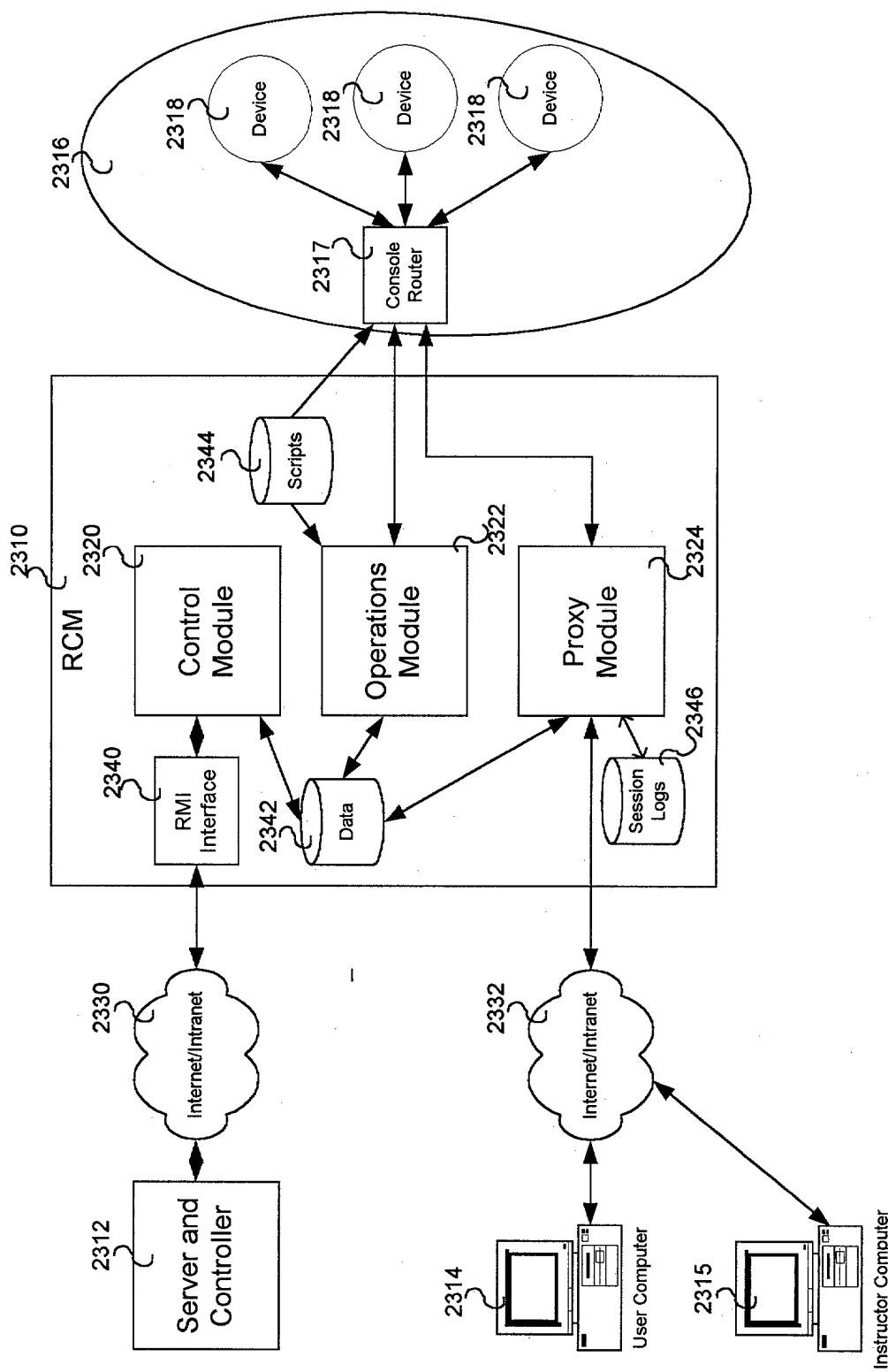


Figure 23

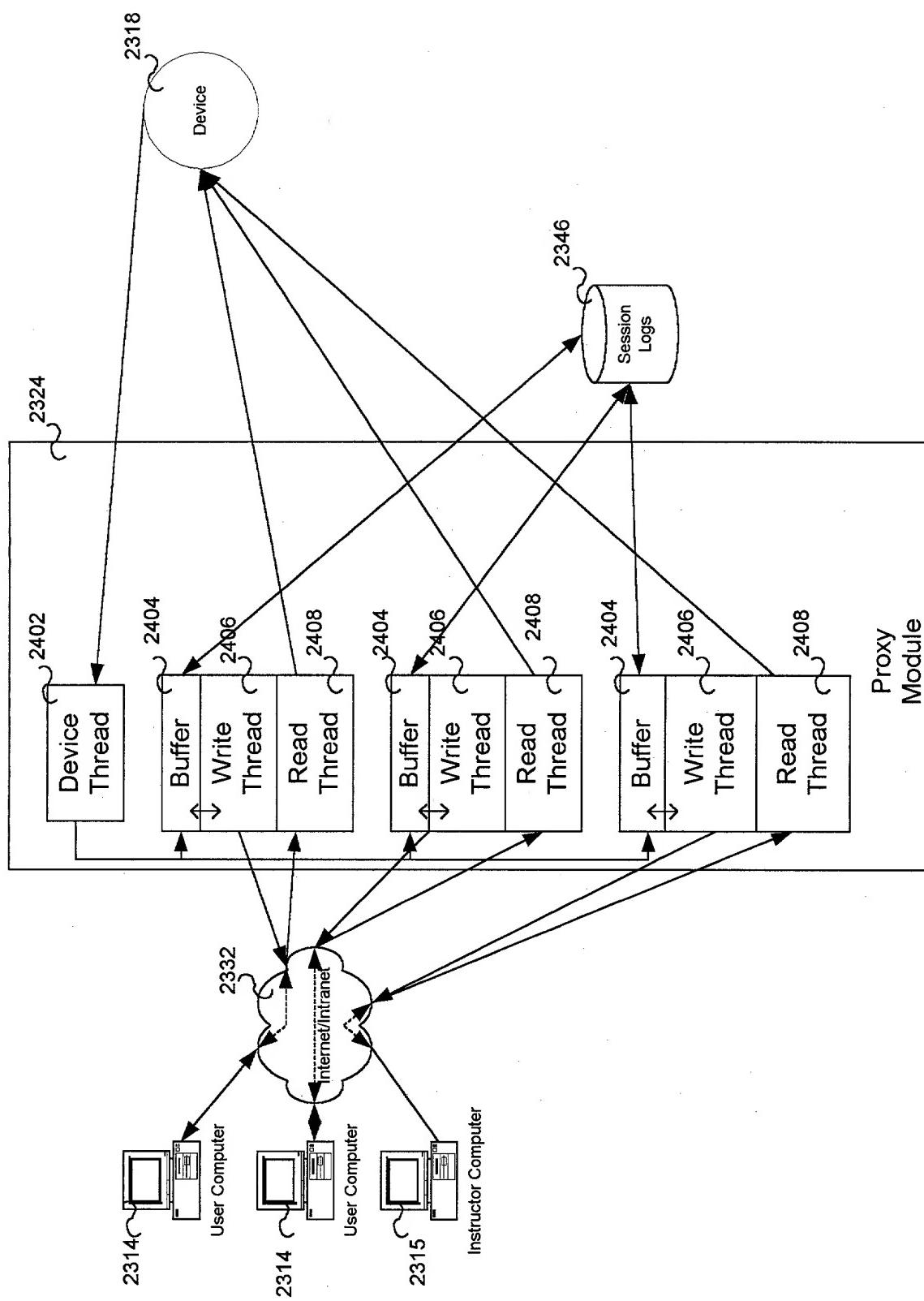


Figure 24